The Movement Complexity of Sorting

Sichen Wang
The Chinese University of Hong Kong, Shenzhen
sichenwang@cuhk.edu.cn

Abstract

The complexity of sorting is usually measured by the number of comparisons. This measure ignores the movement of the read heads between comparisons, which is a real cost on many machines. We study this cost on its own. Keys lie on a d-dimensional grid, two heads move over the grid and compare the keys beneath them, and the only cost is the total movement of the heads. The algorithm plans its route in r rounds and sees the outcomes of each round only after running it. For every d and r we determine the worst-case expected movement needed to sort n keys: it is n^β up to polylogarithmic factors, where $\beta = 1 + D^{r-1}/(D^r - (D-1)^r)$ and $D = 2d$, falling from n^2 at $r = 1$ to $n^{1+1/(2d)}$ with unlimited rounds. Our algorithm makes every round move about the same amount, and a matching lower bound holds against every algorithm.

1 Introduction

The comparison model counts comparisons and nothing else, which for sorting gives the classical $\Theta(n \log n)$. This count is not the whole cost of sorting. On general sequential models the product of time and space for sorting is $\Omega(n^2/\log n)$ [BC82], later strengthened to $\Omega(n^2)$ [Bea91]. On many machines the further cost is travel, because the two keys of a comparison must first be brought within reach. On a Turing machine with one worktape and a two-way read-only input tape, sorting an input of N bits, made of strings of $3 \log N$ bits each, takes $\Omega(N^{3/2}/\sqrt{\log N})$ steps [MS86; DMS91]. This bound comes from the distance the heads must travel: two heads on tapes of length t need $\Omega(t^{3/2})$ steps to reach the pairs of tape positions given by the permutation that transposes a $\sqrt{t} \times \sqrt{t}$ matrix [DMS91].

Adaptivity has its own classical measure. In the parallel comparison model [Val75], an algorithm commits a set of comparisons in each round and learns the outcomes when the round ends, and the exponents of the number of comparisons for sorting, selection, and merging in r rounds are known [Val75; BT83; HH82; AAV86; AA88; Pip87; BB90]. This theory charges nothing for moving between comparisons, and the bounds on travel above say nothing about rounds. How much must a processor travel to sort n keys it cannot move, if it plans its route in r rounds and learns the outcomes of each round only after running it?

We study the simplest model with this cost. The n keys lie one at each position of the grid $[m]^d$, where $n = m^d$, and they carry a hidden total order. Two heads move over the grid in unit steps, and the algorithm learns about the order only by comparing the two keys beneath the heads. It must output the list of positions in increasing order of rank. The positions of the two heads form a point of the *comparison grid* $[m]^{2d}$, and two keys are compared when this point reaches their pair of positions, in either order. An r-round algorithm commits a route at the start of each round, as a function of its coins and of the outcomes of earlier rounds, and it receives the outcomes at all points of the route only after running it. Comparisons, computation, and memory are free, and the only cost is the *movement*, the total ℓ_1 distance the heads move. The algorithm must be correct on every input and for every outcome of its coins, and $M_r^{(d)}(n)$ is the least expected movement on a worst input. For $r = \infty$ the number of rounds is free. Section 2.1 gives the precise definition. The grid is the basic metric space of polynomial growth, and the proofs use little beyond the sizes of its balls and its partitions into cubes and boxes.

Two baselines bound the answer. For $r = 1$, the oblivious case, the single route is fixed in advance and must compare all $\binom{n}{2}$ pairs, so $M_1^{(d)}(n) = \Theta(n^2)$. For $r = \infty$, the fully adaptive case, a merge sort attains $O(n^{1+1/(2d)})$, and this matches the lower bound of Proposition 4.5.

Problem. For every dimension d and every round budget r, determine $M_r^{(d)}(n)$, the least worst-case expected movement with which an r-round algorithm sorts n keys on $[m]^d$.

1.1 Our Results

We determine the exponent for every d and r.

Theorem 1.1 (Movement Exponent). *Fix $d \geq 1$ and $r \in \{1, 2, \dots\} \cup \{\infty\}$, and let $D = 2d$. For finite r put*

$$\beta_r^{(d)} = 1 + \frac{D^{r-1}}{D^r - (D-1)^r},$$

so that $\beta_1^{(d)} = 2$*, and let* $\beta_\infty^{(d)} = \lim_{r\to\infty} \beta_r^{(d)} = 1 + \frac{1}{2d}$*. Then* $M_r^{(d)}(n) = \widetilde{\Theta}\big(n^{\beta_r^{(d)}}\big)$*, and* $M_r^{(d)}(n) = \Theta\big(n^{\beta_r^{(d)}}\big)$ *whenever* $d = 1$*,* $r \le 2$*, or* $r = \infty$*. The upper bounds are attained by explicit polynomial-time algorithms, deterministic for* $r = 1$ *and* $r = \infty$ *and randomized otherwise.*

The exponent decreases from 2 at $r = 1$ to $1 + \frac{1}{2d}$ as $r \to \infty$, each further round lowering it by a geometrically shrinking amount (Remark 2.6), and for $r \ge 2$ it also decreases in d. On the line with unlimited rounds it is $3/2$, the exponent of the head travel on the Turing machine above (Remark 5.4). The number of heads does not affect it: two heads are what a comparison needs, and any fixed number of heads changes the complexity by at most a constant factor (Proposition 7.1). We call the range $d \ge 2$, $3 \le r < \infty$ the *interior*. The logarithmic factors hidden by $\widetilde{\Theta}$ occur only there. Table 1 lists the exponent at small parameters, and Section 6 ends with the proof of the theorem.

Table 1: The movement exponent $\beta_r^{(d)}$. Every entry gives the exact order $\Theta\big(n^{\beta_r^{(d)}}\big)$, except those marked †, which lie in the interior. There the order is known up to a polylogarithmic factor and conjectured exact (Conjecture 8.1).

	$r = 1$	$r = 2$	$r = 3$	$r = 4$	$r = \infty$
$d = 1$	2	$\frac{5}{3}$	$\frac{11}{7}$	$\frac{23}{15}$	$\frac{3}{2}$
$d = 2$	2	$\frac{11}{7}$	$\frac{53}{37}$†	$\frac{239}{175}$†	$\frac{5}{4}$
$d = 3$	2	$\frac{17}{11}$	$\frac{127}{91}$†	$\frac{887}{671}$†	$\frac{7}{6}$

On the line the exponent is tied to selection rather than to sorting (Remark 5.12): $\beta_r^{(1)} = 1 + \frac{1}{2}s_r$, where $s_r = 1 + 1/(2^r - 1)$ is the exponent of the worst-case number of comparisons that a deterministic algorithm needs to find the maximum of n keys in r rounds [Val75], while the number of comparisons for sorting in r rounds has exponent $1 + 1/r$ [AAV86].

1.2 Overview

The cost of a round depends on how many pairs of positions it must visit and on how these pairs are spread. Call an unordered pair of positions a *cell*. If Q cells are spread evenly over the n^2 points of the comparison grid, the distance from each of them to the nearest other one, their *spacing*, is about $(n^2/Q)^{1/D}$, and a route through them moves about $Q \cdot (n^2/Q)^{1/D}$: their number times their spacing, as for tours through random point sets [BHH59; Ste81; Yuk98]. The two baselines are instances of this product. All $\binom{n}{2}$ cells have spacing 1, which gives n^2. The $\Theta(n)$ cells of consecutive ranks, which every algorithm must compare, have spacing about $n^{1/D}$ on a random input, which gives $n^{1+1/(2d)}$.

With r rounds, the early rounds can reduce the number of cells left for the final one, at the cost of their own movement. Our algorithm makes every round move about the same amount. This gives one equation per round, and the solution of these equations is the exponent of Theorem 1.1. The lower bound leads to the same equations, which is why the two exponents agree.

The Algorithm. The algorithm keeps *segments*, sets of keys of consecutive ranks whose internal order is not yet known, and refines them round by round. In round 1 it chooses n/b_0 random positions, the *splitters*, for a parameter b_0. One head stops on each splitter in turn while the other sweeps the grid, so the movement is n^2/b_0, and the keys ranked between two consecutive splitters form a segment of about b_0 keys. The final round, the *cleanup*, compares all pairs inside segments. If

round 2 is the final one, these are about nb_0 cells, known in advance, and a serpentine route through them moves about their number times their spacing (Section 5.1), as in the classical tours through given points [Few55; PB89]. Making the two movements equal gives the exponent for $r = 2$.

With more rounds, refinement rounds come in between. A refinement round selects *pivots*, keys that are compared with every key of their segment, and these cut the segments further. Selecting each key independently, at the rate the refinement needs, fails: the cells of the round are then spread over the whole grid, and the route through them moves a polynomial factor more than the other rounds. So the round selects pivots in groups of nearby positions. It cuts the grid into cubes of side h, selects each cube with probability p, and takes the keys of the selected cubes as pivots. The cells then lie in the *product boxes* $I \times J$ of a selected cube J and another cube I, and the heads go through these boxes one after another. If the segments entering the round have about b keys, the heads move about $p\,h\,nb$ inside the boxes and about $p\,n^{2/d}(nb)^{1-1/d}/h$ between them (Lemmas 5.9 and 6.3), and $h = (n/b)^{1/D}$ makes the two equal.

A segment that meets no selected cube is not cut, so the refined segments can still have many keys, but they meet few cubes. List a key's segment in rank order, starting at the key: each new cube is selected with probability p, so a pivot appears after $1/p$ cubes in expectation. Hence the *dispersion*, the sum over the keys of the number of cubes that the key's segment meets, has expectation about n/p, as for segments of $1/p$ keys. On the line, a bound on the dispersion is all that the next round needs.

The parameters of the algorithm (Algorithm 1) are a decreasing sequence of rank scales $b_j = n^{a_j}$, $0 \le j \le r-2$, the *schedule*, with finest scale $g = b_{r-2}$. On the line, round $j+1$ uses $p = 1/b_j$ and $h = (n/b_{j-1})^{1/D}$, so its output has dispersion about nb_j, as for segments of b_j keys. For $d \ge 2$ the samples are more elaborate, as explained below, but the movement has the same exponent. Requiring every round to move the same amount n^β gives one equation per round:

$$\begin{aligned}
&\text{splitter:} && \beta = 2 - a_0, \\
&\text{refinement } j\text{:} && \beta = 1 + \frac{1-a_{j-1}}{D} + a_{j-1} - a_j \qquad (1 \le j \le r-2), \\
&\text{cleanup:} && \beta = 1 + a_{r-2} + \frac{1-a_{r-2}}{D}.
\end{aligned}$$

The right-hand sides are the exponents of n^2/b_0, of $p\,h\,nb_{j-1}$, and of $ng\,(n/g)^{1/D}$, which is again a number times a spacing. By Lemma 2.5, the unique solution has $\beta = \beta_r^{(d)}$.

For $d \ge 2$ the cleanup needs more. Visiting Q cells in the product box of two cubes of side h moves about $h\,Q^{1-1/D}$. On the line this exponent is $\frac{1}{2}$, and the sum of the $\sqrt{Q}$ over the boxes is at most the dispersion. For $d \ge 2$ the exponent is larger, and the sum of the $Q^{1-1/D}$ also depends on the sizes of the segments. So the cleanup needs small dispersion, for the tour between the product boxes, and a small sum of squared segment sizes, for the visits inside them. Sampling cubes gives the first and sampling keys gives the second, so at $r = 3$ the refinement round draws both samples (Section 6.2). For $r \ge 4$ a refinement round can be followed by another one, whose route moves between boxes of every size, so its cost depends on the dispersion of its input at every box size. One sample controls one size, so each refinement round draws one sample for each box size of a nested partition of the grid, up to a threshold, and the analysis keeps an upper bound on the expected dispersion for each of these sizes (Section 6.3). One factor $\log n$ comes from the number of samples and the other from the number of box sizes at which each route moves.

The Lower Bound. By Yao's principle, it suffices to consider deterministic algorithms on the *uniform input*, a uniformly random assignment of the ranks to the positions [Yao77]. Call a cell *co-blocked at scale* b if its two ranks lie in a common block $\{(i-1)b+1,\dots,ib\}$ for some i. There are $\Theta(ng)$ cells co-blocked at scale g. Take one of them, $\{u,v\}$, such that neither u nor v has been compared in earlier rounds with another position of their block. The earlier outcomes do not determine the order of u and v, because a chain of outcomes from one to the other stays inside the block and starts with a comparison of u or v with another position of the block. So these outcomes are consistent with two inputs that give u and v adjacent ranks, in opposite orders. The final round runs the same route on both, and only the cell $\{u,v\}$ tells them apart, so this route must visit it (Lemma 4.2). On the uniform input a constant fraction of these cells are at distance about $(n/g)^{1/D}$ from all others (Lemma 3.3). So if the earlier rounds compare only a small fraction of the positions inside their blocks, the final route moves about $ng\,(n/g)^{1/D}$ (Fact 3.2), and the cleanup equation says that this is n^β.

It remains to show that rounds moving much less than n^β compare few positions inside their blocks. With the scales rounded to powers of two, every block at scale g lies in a block at each scale b_j, so it suffices that such a round $j+1$ compares few positions inside their blocks at scale b_j. The route of round 1 is fixed before any outcome is known, so each cell on it is co-blocked at scale b_0 with probability at most about b_0/n. A route of movement n^β therefore visits about $n^\beta b_0/n$ such cells, and the splitter equation says that this number is n. So a much shorter route compares only a small fraction of the positions inside their blocks. For $r=2$ we have $b_0=g$, and the proof is complete.

From round 2 on, a route can use what the algorithm has learned to choose cells that are likely to be co-blocked, and this adaptivity is the hard part of the lower bound. The proof handles it round by round, like the chronogram method for dynamic data-structure lower bounds [FS89], with rounds in place of epochs, and like round elimination in communication complexity [MNSW98]. For round $j+1$, two facts replace the estimate above.

The first bounds the number of co-blocked cells that a route can visit. On the uniform input, with high probability, every route of movement L visits at most about $\left(1+L(b_{j-1}/n)^{1/D}\right)\log n$ cells co-blocked at scale b_{j-1} (Lemma 3.4). This is a property of the input alone, so it holds for adaptive routes.

The second bounds the probability that such a cell is co-blocked at the next scale. The proof gives the algorithm more information, which can only help it: after round j, for $j\le r-2$, the index, or *label*, of the block of each position at scale b_{j-1} is revealed at no cost. Given all labels and outcomes so far, the positions of a block that have not been compared with another position of the block still receive the ranks left to them in uniformly random order, because exchanging their ranks changes no label and no outcome. So if at least half of the block is of this kind, a cell that is co-blocked at scale b_{j-1} and contains such a position is co-blocked at scale b_j with conditional probability only $O(b_j/b_{j-1})$ (Lemma 4.8). Blocks with fewer such positions are counted as entirely compared.

Multiplying the two bounds, a route of movement n^β in round $j+1$ visits about $n^\beta(b_{j-1}/n)^{1/D}\,b_j/b_{j-1}$ cells of this kind, up to a logarithmic factor, and the refinement equation says that this number is n. So here too a much shorter route compares only a small fraction of the positions inside their blocks.

Removing the Logarithms. The factors $\log^2 n$ in the upper bound and $\log n$ in the lower bound do not always occur. The factor $\log^2 n$ comes only from the samples for many box sizes, so the upper bound has no logarithmic factor for $d=1$ and for $r\le 3$. The factor $\log n$ comes only from Lemma 3.4, where the largest number of co-blocked cells in one product box enters. For $r\le 2$ the

route of the first round is fixed in advance, so this lemma is not needed. On the line, Lemmas 4.13 and 4.14 avoid the loss: they group nearby co-blocked cells into clusters, a route visits few cells of small clusters per unit of movement, and in expectation at most one cell lies in a large cluster. For $r = \infty$, a merge sort over a partition of the grid into boxes matches the bound $n^{1+1/(2d)}$ given by the cells of consecutive ranks (Propositions 4.5 and 5.3).

Section 2 defines the model and solves the equations, and Section 3 proves the geometric facts used by the lower bound. Section 4 proves the lower bound, with the exact order on the line in Section 4.4, and Sections 5 and 6 prove the upper bounds. Section 7 treats more than two heads, and Section 8 discusses the remaining gap.

1.3 Related Work

The models closest to ours attach costs to comparisons. Priced information charges every query and asks for the cheapest correct computation [CFG+02], and for sorting and selection the costs have been taken to be structured or metric [GK01; GK05]. In the metric model a comparison costs the distance between its two keys. There the costs of the comparisons add up independently, whereas here consecutive comparisons share the heads, so a comparison costs only the move from the previous comparison. As a result, the cost of a round depends on how its comparisons are distributed in the comparison grid, and not only on how many there are. Lemma 5.1 and Fact 3.2 give the upper and the lower bound for a given distribution. The benchmark also differs: these works measure a competitive ratio against the cheapest proof of the order, while we determine the absolute cost on a worst input. For sorting this ratio is polylogarithmic for metric costs [GK05], and also when the keys form two classes and comparisons within a class cost more than comparisons across them [GJ24]. For general costs the question is open. A separate line restricts which comparisons are admissible at all, as in generalized sorting and the 1-∞ cost model [HKK11; KN21; BR16].

Sorting and selection in rounds come from Valiant's parallel comparison model [Val75]. They were studied extensively there [BT83; HH82; AAV86; AA88; Pip87; BB90] and remain active [BMP19]. These results count comparisons, and the place of a comparison plays no role in them. We keep the round structure and replace the cost by movement, and on the line the movement exponent is determined by the exponent s_r of Section 1.1.

A universal tour orders n points once and serves every subset in that order. In the plane, the space-filling curve gives a competitive ratio of $\Theta(\log n)$ [PB89; BG89], and no universal tour has a constant ratio [HKL06; Kra26]. On general metrics, sparse partitions [JLN+05; Fil24] and tree covers [GHR06] give polylogarithmic ratios. The a priori TSP asks the same question in expectation [Jai88]. These problems share with ours the commitment of a route before it is known which visits matter. There the tour is fixed once and judged by its ratio to the best tour for the subset, while our algorithm commits a new route every round and we determine the absolute movement.

Travel has also been studied for element distinctness, on the Turing machine above [Pet08] and on the one-tape machine without an input tape [LO94; BABP03]. Other models charge for moving data. On a mesh-connected computer, records travel along the links to their destinations [TK77], and on a VLSI chip, the information that must cross the chip gives area-time trade-offs [Tho83]. Token swapping rearranges labeled tokens along the edges of a graph and counts the swaps [YDI+15]. External-memory sorting transports records between memory levels, and its classical lower bound treats the records as indivisible [AV88]. Under the conjecture of Li and Li that, in undirected networks with several source-sink pairs, network coding achieves no higher throughput than routing [LL04], the same bound holds without this assumption [FHLS19]. Here the keys stay in their positions, so

the lower bound needs neither assumption, and the comparisons it counts are forced by correctness alone (Remark 4.11).

2 Preliminaries

This section makes the model of Section 1 precise and solves the equations of Section 1.2, which give the exponent and the rank scales used by both bounds.

2.1 The Model

Definition 2.1 (The Movement Model)**.** Fix $d \geq 1$ and $m \geq 2$, let $n = m^d$, and write $[m] = \{1, \dots, m\}$.

1. *Input and output.* Every position $x \in [m]^d$ holds one key, and the keys carry a hidden total order: the input is a bijection $\pi\colon [m]^d \to [n]$, and $\mathrm{rk}_\pi(x) = \pi(x)$ is the rank of the key at x. The algorithm learns about π only through the outcomes of item 3. The output is the list of positions in increasing order of rank.

2. *Configurations and routes.* Two heads occupy positions $x, y \in [m]^d$, and the pair (x, y) is their *configuration*. The configurations are the points of the *comparison grid* $[m]^D$, where $D = 2d$. A *route* is a sequence of configurations $z_0, \dots, z_L$ with $\|z_{t+1} - z_t\|_1 = 1$ for $0 \leq t < L$. The *movement* of the route is L. Both heads start at the position $(1, \dots, 1)$.

3. *Comparisons.* The *outcome* at a configuration (x, y) is whether $\mathrm{rk}_\pi(x) < \mathrm{rk}_\pi(y)$. Comparisons, computation, and memory are free.

4. *Rounds.* An *r-round algorithm* proceeds in r rounds. At the start of each round it commits, as a function of its random coins and of the outcomes received in earlier rounds, a route that starts at the current configuration. It then runs the route and receives the outcomes at all configurations on it. No choice made within a round depends on outcomes of that round.

5. *Correctness and cost.* The algorithm must output the correct order for every input and every outcome of its coins. We write $\mathrm{Move}(\mathrm{ALG}, \pi)$ for the total movement of an algorithm ALG on input π, and
$$M_r^{(d)}(n) = \inf_{\mathrm{ALG}} \max_{\pi} \mathbb{E}\big[\mathrm{Move}(\mathrm{ALG}, \pi)\big],$$
the infimum over r-round algorithms. For $r = \infty$, $M_\infty^{(d)}(n) = \inf_r M_r^{(d)}(n)$.

Definition 2.2 (Cells)**.** A *cell* is an unordered pair $\{x, y\}$ of distinct positions. Its two *orientations* are the configurations (x, y) and (y, x). On cells, the metric dist is
$$\mathrm{dist}\big(\{x, y\}, \{x', y'\}\big) = \min\big\{\|x - x'\|_1 + \|y - y'\|_1,\ \|x - y'\|_1 + \|y - x'\|_1\big\},$$
and for a cell w and a set W of cells we put $\mathrm{dist}(w, W) = \min_{w' \in W} \mathrm{dist}(w, w')$, with $\mathrm{dist}(w, \emptyset) = +\infty$.

Observation 2.3. (a) *A route of movement L visits at most $L + 1$ configurations.*

(b) *The two orientations of a cell carry the same information, and a configuration (x, x) carries none. On the configurations with $x \neq y$, the map $(x, y) \mapsto \{x, y\}$ is 1-Lipschitz from ℓ_1 to* dist*. Hence deleting the configurations (x, x) from a route of movement L leaves a sequence of cells whose* dist*-length, the sum of the* dist *distances between consecutive cells, is at most L. The number of cells within* dist*-distance ρ of a cell $\{x, y\}$ is at most the number of points*

of $[m]^D$ *within* ℓ_1*-distance* ρ *of* (x, y)*. On the line,* $(x, y) \mapsto (\min\{x, y\}, \max\{x, y\})$ *folds the configurations onto* $\{x \le y\}$ *and the cells onto* $\{x < y\}$*, and* dist *is the* ℓ_1 *distance between the folded cells.*

Proof. Part (a) holds because the route consists of $z_0, \ldots, z_L$. For (b), when $x \neq y$ the outcome at (y, x) is the negation of the outcome at (x, y), and the outcome at (x, x) is always negative. By definition, $\mathrm{dist}(\{x, y\}, \{x', y'\})$ is the smaller of the ℓ_1 distances from (x, y) to the two orientations of $\{x', y'\}$, so the ℓ_1 distance of two configurations is at least the dist of their cells, which is the Lipschitz property. Since the swap $(x, y) \mapsto (y, x)$ is an isometry of ℓ_1, dist is a metric on cells. Between two consecutive configurations that survive the deletion, the route moves at least their ℓ_1 distance, so the cells have dist-length at most L. A cell within dist-distance ρ of $\{x, y\}$ has an orientation within ℓ_1-distance ρ of (x, y), and distinct cells have no orientation in common, which gives the count. On the line, if $x < y$ and $x' < y'$, then $t \mapsto |t - y'| - |t - x'|$ is nonincreasing, and comparing its values at x and at y gives $|x - x'| + |y - y'| \le |x - y'| + |y - x'|$. So the minimum in dist is the ℓ_1 distance between (x, y) and (x', y'). □

Since the keys do not move, we identify each key with its position, so a set of keys is a set of positions. When π is fixed we write rk for rk_π. The algorithm *compares* x and y in round t if the route of round t visits an orientation of $\{x, y\}$. In the same way, a route visits a set of cells if it visits an orientation of each, and a cell lies in a set of configurations if one of its orientations does.

Notation. Throughout, d and r are fixed and asymptotic statements refer to $n \to \infty$. A subscript on an asymptotic symbol lists the parameters other than n on which the hidden constant depends, in the order d, r, and, in Section 7, k, as in O_d or $\Omega_{d,r}$. A symbol without a subscript hides an absolute constant. Theorem 1.1 carries no subscripts, and its constants depend on d and r. A tilde, as in $\widetilde{\Theta}$, also hides factors polylogarithmic in n. Logarithms are to base 2. When the algorithm and the input are clear, we write Move for $\mathrm{Move}(\mathrm{ALG}, \pi)$.

A *box* is a product of intervals of integers, viewed as a set of lattice points. Its *side* along an axis is the number of its points on a line parallel to that axis, and a *cube* is a box whose sides are equal. A *ball* is the set of grid points within a given ℓ_1 distance of a center. When we partition the line into intervals of h positions, or $[m]^d$ into cubes of side H, the last piece along each axis may be shorter, which affects only constants.

2.2 The Scale Schedule

Definition 2.4 (Scales and Blocks)**.** A *scale* is a number b with $1 \le b \le n$. For an integer scale b, the *scale-b blocks* are the sets of ranks $\{(i-1)b + 1, \ldots, ib\} \cap [n]$ for $i = 1, \ldots, \lceil n/b \rceil$. Each holds b ranks except possibly the last, and i is the *label* of the i-th block. A position lies in a block if its rank does. The scale b is *dyadic* if it is a power of two. Every block of a dyadic scale lies in a block of each larger dyadic scale. For finite $r \ge 2$, a *schedule* is a sequence $b_j = n^{a_j}$, $0 \le j \le r - 2$, with exponents $1 > a_0 > a_1 > \cdots > a_{r-2} > 0$. Its finest term is $g = b_{r-2}$.

For finite $r \ge 2$, both bounds use the schedule determined by the following system in the unknowns $\beta, a_0, \ldots, a_{r-2}$. It has one equation for the splitter round of the algorithm, one for each refinement round $j = 1, \ldots, r - 2$, and one for the cleanup round. Each states that the movement of that round

is n^β, and Section 1.2 explains the three expressions.

$$\text{splitter:} \qquad \beta = 2 - a_0, \tag{2.1}$$

$$\text{refinement } j\text{:} \qquad \beta = 1 + \frac{1 - a_{j-1}}{D} + a_{j-1} - a_j \qquad (1 \le j \le r-2), \tag{2.2}$$

$$\text{cleanup:} \qquad \beta = 1 + a_{r-2} + \frac{1 - a_{r-2}}{D}. \tag{2.3}$$

Lemma 2.5. *Let $r \ge 2$ and let $q = (D-1)/D$. The system of Equations (2.1) to (2.3) has the unique solution*

$$\beta = \beta_r^{(d)} = 1 + \frac{1-q}{1-q^r}, \qquad a_j = \frac{q^{j+1} - q^r}{1 - q^r} \quad (0 \le j \le r-2),$$

and $1 > a_0 > a_1 > \cdots > a_{r-2} > 0$, so the a_j define a schedule. We write $\delta = \beta_r^{(d)} - 1 = (1-q)/(1-q^r)$, so $a_0 = 1 - \delta$.

Proof. For a solution, write $\delta = \beta - 1$ and $t_j = 1 - a_j$. Equation (2.1) gives $t_0 = \delta$. Since $q = 1 - 1/D$, Equation (2.2) reads $t_j = q\, t_{j-1} + \delta$, so

$$t_j = \delta\,(1 + q + \cdots + q^j) \qquad (0 \le j \le r-2),$$

and Equations (2.1) and (2.2) determine every a_j from δ. Equation (2.3) reads $\delta = 1 - q\, t_{r-2}$, that is, $\delta\,(1 + q + \cdots + q^{r-1}) = 1$, whose unique solution is $\delta = (1-q)/(1-q^r)$. Hence $\beta = \beta_r^{(d)}$, and $t_j = (1 - q^{j+1})/(1-q^r)$ gives $a_j = (q^{j+1} - q^r)/(1-q^r)$. Finally, $a_j - a_{j+1} = q^{j+1}(1-q)/(1-q^r) > 0$ for $0 \le j \le r-3$, $a_{r-2} = q^{r-1}(1-q)/(1-q^r) > 0$, and $a_0 = 1 - \delta < 1$. □

From now on, for given d and finite $r \ge 2$, the exponents $a_0, \ldots, a_{r-2}$ are those of Lemma 2.5, $b_j = n^{a_j}$ is their schedule, and $g = b_{r-2}$. The algorithms use the b_j as real parameters. In Section 4, where blocks must nest, b_j denotes the largest power of two not exceeding n^{a_j}, which differs from n^{a_j} by a factor less than 2.

Remark 2.6. Since $(1-q)/(1-q^r) = 1/(1 + q + \cdots + q^{r-1})$, and the sum $1 + q + \cdots + q^{r-1}$ increases strictly in r and, for $r \ge 2$, in $q = 1 - 1/(2d)$, the exponent $\beta_r^{(d)}$ decreases strictly in r and, for $r \ge 2$, in d. It lies in $(1 + \frac{1}{2d}, 2]$. Its distance to the fully adaptive exponent satisfies

$$\beta_r^{(d)} - \left(1 + \frac{1}{2d}\right) = \frac{q^r(1-q)}{1-q^r}.$$

3 Geometry of the Comparison Grid

This section proves three facts about the comparison grid that the lower bound uses. A route that visits many cells far apart from one another has large movement (Fact 3.2). The other two facts concern the *uniform input*, a uniformly random bijection $\pi\colon [m]^d \to [n]$. With probability at least $7/8$, at least half of the cells whose two positions lie in a common block of ranks are far from all other such cells (Lemma 3.3). With probability at least $1 - n^{-10}$, every route, however it is chosen, visits few such cells per unit of movement (Lemma 3.4).

Definition 3.1 (Co-Blocked Cells)**.** Two positions are *co-blocked* at an integer scale b if their ranks lie in the same scale-b block. A cell is co-blocked at scale b if its two positions are. We write $\mathcal{C}_b$ for the set of cells co-blocked at scale b.

The set $\mathcal{C}_b$ depends on the input, but its size does not. For integer b with $2 \le b \le n$ we have $\frac{1}{4}n(b-1) \le |\mathcal{C}_b| \le \frac{1}{2}n(b-1)$, and in particular $|\mathcal{C}_b| \ge nb/8$. Indeed, $2|\mathcal{C}_b|$ is the sum over all positions x of the number of other positions co-blocked with x. Every position is co-blocked with at most $b-1$ others, which gives the upper bound. At least $n/2$ positions lie in blocks of exactly b ranks, since $b\lfloor n/b \rfloor \ge n/2$, and each of them is co-blocked with exactly $b-1$ others, which gives the lower bound. On the uniform input, two fixed positions are co-blocked at scale b with probability at most b/n: given the rank of one, the rank of the other is uniform over the remaining $n-1$ ranks, at most $b-1$ of which lie in its block, and $(b-1)/(n-1) \le b/n$.

Fact 3.2. *Let Y be a metric space with metric dist_Y, and let $W \subseteq Y$ be finite with $|W| \ge 2$. For $w \in W$ let $\nu_W(w) = \min_{w' \in W \setminus \{w\}} \mathrm{dist}_Y(w, w')$. Then every finite sequence of points of Y that contains every point of W has length at least $\frac{1}{2}\sum_{w \in W} \nu_W(w)$, where the length of a sequence is the sum of the distances between consecutive points.*

Proof. Keeping only the first occurrence of each point of W gives a path through W that, by the triangle inequality, is no longer than the sequence. So the sequence is at least as long as a minimum spanning tree of W. Charge each $w \in W$ to a shortest edge of this tree at w. That edge has length at least $\nu_W(w)$, and each edge is charged at most twice, so $\sum_{w \in W} \nu_W(w)$ is at most twice the length of the tree. □

We apply Fact 3.2 to cells with the metric dist. Let $\rho > 0$, and let a route of movement L visit every cell of a set W of at least two cells that are pairwise more than ρ apart. Deleting the configurations (x, x) from the route leaves a sequence of cells of dist-length at most L that contains every cell of W (Observation 2.3(b)). Since $\nu_W(w) > \rho$ for every $w \in W$, Fact 3.2 gives $L \ge \frac{1}{2}|W|\rho$.

Lemma 3.3. *For every integer scale $b \ge 2$ there is a radius $\rho = \Theta_d\big((n/b)^{1/D}\big)$ such that, with probability at least $7/8$ over the uniform input, at least half of the cells $w \in \mathcal{C}_b$ satisfy $\mathrm{dist}(w, \mathcal{C}_b \setminus \{w\}) > \rho$.*

Proof. We bound the expected number of pairs of cells of $\mathcal{C}_b$ that are close to each other and apply Markov's inequality. Let $\rho = \varepsilon_0 (n/b)^{1/D}$, where $0 < \varepsilon_0 \le 1$ depends only on d and is chosen below. Distinct cells are at distance at least 1, so if $\rho < 1$ every cell satisfies the condition. Assume therefore that $\rho \ge 1$. Let Z be the number of ordered pairs (w, w') of distinct cells of $\mathcal{C}_b$ with $\mathrm{dist}(w, w') \le \rho$. A cell $w \in \mathcal{C}_b$ with $\mathrm{dist}(w, \mathcal{C}_b \setminus \{w\}) \le \rho$ is the first entry of such a pair, so at most Z cells violate the condition. It therefore suffices to show that $\mathbb{E}Z \le \frac{1}{16}|\mathcal{C}_b|$: then Markov's inequality applied to Z gives $Z \le \frac{1}{2}|\mathcal{C}_b|$ with probability at least $7/8$.

A ball of radius ρ in $[m]^d$ contains at most $(2\rho+1)^d \le (3\rho)^d$ positions, and by Observation 2.3(b) at most $(3\rho)^D$ cells lie within distance ρ of a given cell. Two distinct cells share at most one position, so every pair counted by Z falls into exactly one of the following two cases.

- The cells share a position: $w = \{u, v\}$ and $w' = \{u, v'\}$ with $v \ne v'$. Both lie in $\mathcal{C}_b$ only if the ranks of u, v, and v' lie in one block. Given the rank of u, this has probability at most $\frac{b-1}{n-1} \cdot \frac{b-2}{n-2} \le (b/n)^2$. The condition $\mathrm{dist}(w, w') \le \rho$ requires $\|v - v'\|_1 \le \rho$, or $\|u - v'\|_1 + \|v - u\|_1 \le \rho$, in which case both v and v' lie within distance ρ of u. There are therefore at most $n^2(3\rho)^d + n(3\rho)^{2d}$ such triples (u, v, v'), and these pairs contribute at most

$$\Big(\frac{b}{n}\Big)^2 \big(n^2(3\rho)^d + n(3\rho)^{2d}\big) = 3^d \varepsilon_0^d \, nb \Big(\frac{b}{n}\Big)^{1/2} + 9^d \varepsilon_0^{2d} \, b \le 2 \cdot 9^d \varepsilon_0^d \, nb.$$

- The cells have four distinct positions. There are at most $\frac{1}{2}n^2(3\rho)^D$ such pairs. For each of them both cells lie in $\mathcal{C}_b$ with probability $O(b^2/n^2)$, because the first cell is co-blocked with probability at most b/n and, given the ranks of its positions, the second with probability at most $(b-1)/(n-3)$. These pairs contribute

$$O\Big(\frac{b^2}{n^2}\Big)\cdot n^2(3\rho)^D = O\big(9^d\varepsilon_0^{2d}\,nb\big).$$

Since $|\mathcal{C}_b| \geq nb/8$ and $\varepsilon_0 \leq 1$, there is a constant c_d depending only on d such that the two contributions give $\mathbb{E}Z \leq c_d\,\varepsilon_0^d\,|\mathcal{C}_b|$. Choosing ε_0 with $c_d\varepsilon_0^d \leq \frac{1}{16}$ completes the proof. □

Lemma 3.4. *Fix a constant $\alpha < 1$. There is a constant c depending only on d and α with the following property. For a dyadic scale $b \leq n^\alpha$, let E_b be the event that every route visits at most $c\big(1 + L(b/n)^{1/D}\big)\log n$ cells of $\mathcal{C}_b$, where L is the movement of the route. Then, for sufficiently large n, with probability at least $1 - n^{-10}$ over the uniform input, E_b holds for every dyadic scale $b \leq n^\alpha$.*

Proof. We partition the comparison grid into products of two cubes, show that with high probability none of these products contains many cells of $\mathcal{C}_b$, and then cut a route into pieces that each stay in few of them. Let n be large enough for the estimates below, and fix a dyadic scale $b \leq n^\alpha$.

Let $s = \lceil (n/b)^{1/D} \rceil$ and partition $[m]^d$ into cubes of side s, clipped at the boundary. Since $(n/b)^{1/D} \geq 1$, each cube contains at most $s^d \leq V$ positions, where $V = 2^d(n/b)^{1/2}$. A *product box* is a set $I \times J \subseteq [m]^D$ for two of these cubes I and J. There are at most n^2 product boxes. The *load* of $I \times J$ is $|\mathcal{C}_b \cap (I \times J)|$, the number of cells of $\mathcal{C}_b$ that lie in $I \times J$. Cells that share a position are correlated, while disjoint cells are nearly independent, so we show that a product box of large load contains either many disjoint cells or a block with many of its positions. Let $\zeta = \lceil 30/(1-\alpha) \rceil$, $\mu = \lceil \max\{12 \cdot 4^d,\, 13\log n\} \rceil$, and $\Lambda_0 = 2\zeta\mu = O_{d,\alpha}(\log n)$. We show that a fixed product box $I \times J$ has load at least Λ_0 with probability at most $2n^{-13}$. View the cells counted by the load as the edges of a graph on $I \cup J$. Every edge joins two positions of one block. If every block contains fewer than ζ positions of $I \cup J$, then every vertex has degree less than ζ, and a greedy matching, which discards fewer than 2ζ edges for each edge it keeps, finds μ disjoint edges among any $\Lambda_0 = 2\zeta\mu$ edges. Hence, if the load is at least Λ_0, one of the following two events occurs.

- Some block contains at least ζ positions of $I \cup J$. The probability that ζ given positions all lie in a given block is at most $(b/n)^\zeta$, and $I \cup J$ has at most $2V$ positions. A union bound over the at most n blocks and the at most $(2V)^\zeta$ sets of ζ positions bounds the probability of this event by

$$n\Big(\frac{2Vb}{n}\Big)^\zeta \leq n\big(2^{d+1}n^{-(1-\alpha)/2}\big)^\zeta \leq n^{-13},$$

  using $b \leq n^\alpha$ and $(1-\alpha)\zeta/2 \geq 15$.

- Some μ disjoint cells of $\mathcal{C}_b$ lie in $I \times J$. At most V^2 cells lie in $I \times J$, so there are at most $(V^2)^\mu/\mu!$ sets of μ disjoint cells in $I \times J$. Given the ranks of the positions of the earlier cells of such a set, the next cell is co-blocked with probability at most $(b-1)/(n-2\mu) \leq 2b/n$, so all its cells are co-blocked with probability at most $(2b/n)^\mu$. Since $V^2 b/n = 4^d$ and $\mu! \geq (\mu/3)^\mu$, a union bound over these sets bounds the probability of this event by

$$\frac{(2\cdot 4^d)^\mu}{\mu!} \leq \Big(\frac{6\cdot 4^d}{\mu}\Big)^\mu \leq 2^{-\mu} \leq n^{-13}.$$

A union bound over the at most $\log n + 1$ dyadic scales $b \le n^\alpha$ and the at most n^2 product boxes of each shows that, with probability at least $1 - n^{-10}$, at every dyadic scale $b \le n^\alpha$ every product box has load less than Λ_0.

On this event we bound the number of cells of $\mathcal{C}_b$ that a route visits. Cut a route $z_0, \dots, z_L$ into the pieces $z_{is}, \dots, z_{\min\{(i+1)s, L\}}$ for $0 \le i \le \lfloor L/s \rfloor$. There are at most $1 + L/s \le 1 + L(b/n)^{1/D}$ of them. Every configuration (x', y') of a piece lies within ℓ_1-distance s of the first configuration (x, y) of the piece, so x' lies in the cube of side $2s+1$ centered at x, and y' in the cube of side $2s+1$ centered at y. Along each axis, $2s+1$ consecutive integers meet at most three intervals of the partition, so each of these two cubes meets at most 3^d cubes of the partition, and the piece stays in at most 9^d product boxes. Every cell that the piece visits lies in one of them, so the piece visits fewer than $9^d\Lambda_0$ cells of $\mathcal{C}_b$, and the route visits fewer than $9^d\Lambda_0\big(1 + L(b/n)^{1/D}\big)$. Since $\Lambda_0 = O_{d,\alpha}(\log n)$, this is the bound of E_b for a suitable constant c depending only on d and α. □

The logarithm in Lemma 3.4 comes from the largest load over all product boxes. The expected load of a fixed product box is at most 4^d, since at most $4^d n/b$ cells lie in it, each co-blocked with probability at most b/n.

4 The Lower Bound

This section proves the lower-bound half of Theorem 1.1.

Theorem 4.1 (Lower Bound)**.** *For every fixed $d \ge 1$ and $r \ge 1$, $M_r^{(d)}(n) = \Omega_{d,r}\big(n^{\beta_r^{(d)}} / \log n\big)$.*

The proof has the same plan for every finite $r \ge 2$. Correctness forces the final round to visit every cell that is co-blocked at the finest scale and whose positions the earlier rounds have not compared inside their block (Section 4.1). Unless an earlier round already has movement about $n^{\beta_r^{(d)}}$, up to a logarithmic factor, many such cells remain and lie far apart, so the final round has this movement. For $r \le 2$ this argument is short and loses no logarithmic factor. Section 4.2 gives it, together with the bound $\Omega_d(n^{1+1/(2d)})$ for every r, including $r = \infty$. For $r \ge 3$ the middle rounds are adaptive, and Section 4.3 controls what they can learn. Section 4.4 removes the logarithmic factor on the line.

The lower bounds are proved on the uniform input. Fixing the coins of an r-round algorithm that is correct on every input and for every outcome of its coins gives a deterministic r-round algorithm that is correct on every input. Hence, by Yao's principle [Yao77], a lower bound on the expected movement on the uniform input that holds for every deterministic r-round algorithm correct on every input is a lower bound on $M_r^{(d)}(n)$. The lemmas of Section 4.1 hold for every input, and the uniform input is used only when we count.

4.1 The Augmented Model and Forced Comparisons

For finite $r \ge 2$, let $\beta = \beta_r^{(d)}$. Recall from Section 2.2 that in this section b_j is the largest power of two not exceeding n^{a_j}, where $a_0 > \cdots > a_{r-2}$ are the exponents of Lemma 2.5, and that $g = b_{r-2}$. Thus $n^{a_j}/2 < b_j \le n^{a_j}$. The scales b_j are powers of two, so every scale-b_j block lies in a scale-b_i block for $i < j$. Since $a_{r-2} > 0$ and $a_{j-1} > a_j$, for n larger than a constant depending on d and r we have $g \ge 2$ and $b_{j-1} \ge 2b_j$ for $1 \le j \le r-2$, and we assume this throughout the section.

A scale-b block is *full* if it holds b ranks. Otherwise it is *partial*. Every b_j divides b_0, so the positions whose rank lies in a partial block of some scale b_j are exactly those of rank greater than $b_0\lfloor n/b_0 \rfloor$,

and there are fewer than b_0 of them. The proofs below set these positions aside, because their counts need blocks of exactly b ranks.

From round 2 on, a route can use earlier outcomes to choose cells that are likely to be co-blocked, so the probability b/n no longer applies to the cells it visits. We therefore give the algorithm more information, which can only help it, in a form that leaves, inside most blocks, the ranks of the positions not yet compared in the block uniformly random (Lemma 4.8). In the *augmented model*, after round j, for $1 \le j \le r-2$, the label of the scale-b_{j-1} block of every position is revealed at no cost, and the route of each later round may depend on the revealed labels. Thus the route of round 1 depends on no outcome and no label, and for $j \ge 1$ the route of round $j+1$ may depend on the outcomes of rounds $1, \dots, j$ and on the labels revealed after them. The *revealed scales* are $b_0, \dots, b_{r-3}$, and the labels at the finest scale g are never revealed. An algorithm of the original model is an algorithm of the augmented model that ignores the labels, so a lower bound in the augmented model is a lower bound in the original one. Once the labels at a scale are revealed, a comparison of two positions in different blocks of that scale has a known outcome, so all further information about the ranks comes from comparisons inside these blocks.

We now show which comparisons correctness forces. Let P_t be the partial order on positions generated by the relations $x < y$, where $\{x, y\}$ ranges over the cells compared in rounds $1, \dots, t$ and $\mathrm{rk}(x) < \mathrm{rk}(y)$. Revealed labels separate only positions in different blocks, so they do not help to order two positions that have the same labels:

Lemma 4.2. *Let a deterministic r-round algorithm in the augmented model be correct on every input, and fix an input. If two positions u and v are incomparable in P_{r-1} and have the same label at every revealed scale, then the route of round r visits the cell $\{u, v\}$.*

Proof. We construct two inputs that agree with the given input on all revealed labels and on all outcomes of rounds $1, \dots, r-1$, but order u and v oppositely. Correctness then forces round r to compare u and v.

If $r \le 2$, nothing is revealed, and we let $\lambda(x) = 1$ for every position x. If $r \ge 3$, let $\lambda(x)$ be the label of x at the finest revealed scale b_{r-3}. The revealed partitions nest, so $\lambda(x)$ determines the labels of x at all revealed scales. In both cases $\lambda(u) = \lambda(v)$. Let P' be the partial order generated by P_{r-1} and the relations $x < y$ for all positions x, y with $\lambda(x) < \lambda(y)$. The given input extends P', because blocks are intervals of ranks ordered by their labels.

The positions u and v are incomparable in P'. Indeed, along a relation of P_{r-1} the label λ does not decrease, because the two positions of a compared cell are ordered by rank and every block is an interval of ranks. Along an added relation, λ increases strictly. Hence λ does not decrease along a chain of P' from u to v, and since $\lambda(u) = \lambda(v)$, it is constant on the chain. So the chain uses no added relation and lies in P_{r-1}, which contradicts the incomparability of u and v in P_{r-1}. The same argument rules out a chain from v to u.

We now build the two inputs. List first the positions that lie below u or v in P', in an order that extends P'; then u and v, in either order; then the remaining positions, in an order that extends P'. No remaining position lies below a position of the first part, and since u and v are incomparable in P', neither of them lies below a position of the first part. So both lists extend P'. Read as rank assignments, they are two inputs π_1 and π_2 that differ by exchanging two adjacent ranks at u and v. For $r \ge 3$, both have the given labels at every revealed scale: in a linear extension of P' the positions with $\lambda = i$ receive consecutive ranks, and there are as many of them as the block with label i has ranks, so they receive exactly those ranks.

Finally we compare the executions. Since π_1 and π_2 extend P_{r-1} and have the given labels, induction over the rounds shows that on both inputs the algorithm runs the same routes in rounds $1, \dots, r-1$ as on the given input and receives the same outcomes, so it also runs the same route in round r. A comparison other than $\{u, v\}$ has the same outcome on π_1 and π_2: if it involves neither u nor v, its two ranks are unchanged, and if it involves u or v and a third position, the rank of the third position lies on the same side of the two adjacent ranks. If the route of round r did not visit $\{u, v\}$, all outcomes on π_1 and π_2 would agree, the algorithm would output the same list on both inputs, and it would be wrong on one of them. □

Two positions of one block can be ordered only through a comparison inside the block:

Lemma 4.3. *Let P be the partial order generated, in the same way as P_t, by the outcomes of a set of comparisons. If $u <_P v$ and u and v lie in a common scale-b block, then one of these comparisons involves u and another position of this block.*

Proof. Write $u <_P v$ as a chain $u = x_0, x_1, \dots, x_i = v$ in which every two consecutive positions form a compared cell and have increasing ranks. The ranks along the chain lie between $\mathrm{rk}(u)$ and $\mathrm{rk}(v)$, hence in the block. So the first cell of the chain, $\{u, x_1\}$, is a comparison of u with another position of the block. □

Together, Lemmas 4.2 and 4.3 show that round r must visit every cell co-blocked at scale g whose two positions have not been compared, in rounds $1, \dots, r-1$, with other positions of their scale-g block. Indeed, by Lemma 4.3 the two positions are incomparable in P_{r-1}, and since every revealed scale is coarser than g, they have the same revealed labels.

4.2 Two Rounds and the Fully Adaptive Bound

The proof for $r = 2$ has four stages, which the proof for $r \geq 3$ in Section 4.3 follows. The route of round 1 is fixed before any outcome is known, so there is no adaptivity to control, Lemma 3.4 is not needed, and no logarithmic factor is lost.

Proposition 4.4. *For every fixed d and every $r \leq 2$, $M_r^{(d)}(n) = \Omega_d\big(n^{\beta_r^{(d)}}\big)$.*

Proof. By Yao's principle, fix a deterministic r-round algorithm that is correct on every input and let π be uniform. We show that $\mathbb{E}[\mathrm{Move}] = \Omega_d\big(n^{\beta_r^{(d)}}\big)$.

For $r = 1$, the order P_0 is empty and no label is revealed, so by Lemma 4.2 the route visits every cell. By Observation 2.3(a), $\mathrm{Move} \geq \binom{n}{2} - 1 = \Omega(n^2)$, and $\beta_1^{(d)} = 2$.

Now let $r = 2$, so that $g = b_0$ and, by Equation (2.1), $\beta = 2 - a_0$. Write L_1 and L_2 for the movements of the two routes. No outcome is known before round 1, so the route of round 1 and L_1 do not depend on π. Let $\varepsilon = 1/192$. If $L_1 \geq \varepsilon n^\beta$, then $\mathrm{Move} \geq \varepsilon n^\beta$ and there is nothing to prove, so assume $L_1 < \varepsilon n^\beta$. Call a position *clean* if its rank lies in a full scale-g block and it lies in no cell that round 1 visits and that is co-blocked at scale g.

Exposure. The route of round 1 visits at most $L_1 + 1$ cells (Observation 2.3(a)), each co-blocked at scale g with probability at most g/n, and fewer than g positions lie in a partial block. So the expected number of positions that are not clean is less than $g + 2(L_1 + 1)g/n \leq n^{a_0} + 2\varepsilon n^{\beta + a_0 - 1} + 2$. Since $\beta + a_0 - 1 = 1$, this is $n/96 + o(n)$, which is at most $n/64$ for large n.

Survivors. Every position lies in fewer than g cells of $\mathcal{C}_g$, so the expected number of cells of $\mathcal{C}_g$ with a position that is not clean is at most $gn/64 \le |\mathcal{C}_g|/8$. By Markov's inequality, with probability at least $1/2$ at most $|\mathcal{C}_g|/4$ cells of $\mathcal{C}_g$ have such a position. By Lemma 3.3 at scale g, with probability at least $7/8$ at least half of the cells $w \in \mathcal{C}_g$ satisfy $\mathrm{dist}(w, \mathcal{C}_g \setminus \{w\}) > \rho$, where $\rho = \Theta_d\big((n/g)^{1/D}\big)$. On the intersection of the two events, which has probability at least $1 - \frac{1}{2} - \frac{1}{8} = \frac{3}{8}$, let $\mathcal{H}$ be the set of cells of $\mathcal{C}_g$ that have two clean positions and satisfy this condition. Then $|\mathcal{H}| \ge |\mathcal{C}_g|/4 \ge ng/32$, and any two cells of $\mathcal{H}$ are more than ρ apart.

Forcing. Let $\{u, v\} \in \mathcal{H}$. Since u and v are clean, neither has been compared in round 1 with another position of its scale-g block, so by Lemma 4.3 they are incomparable in P_1. No label is revealed, so by Lemma 4.2 the route of round 2 visits $\{u, v\}$.

Cost of the Final Round. Since $|\mathcal{H}| \ge 2$, Observation 2.3(b) and Fact 3.2 give

$$L_2 \;\ge\; \tfrac{1}{2}|\mathcal{H}|\rho \;=\; \Omega_d\big(ng\,(n/g)^{1/D}\big) \;=\; \Omega_d\big(n^{1+a_0+(1-a_0)/D}\big) \;=\; \Omega_d(n^{\beta}),$$

where the second step uses $g > n^{a_0}/2$ and the last is Equation (2.3) for $r = 2$. So, when $L_1 < \varepsilon n^{\beta}$, we have $L_2 = \Omega_d(n^{\beta})$ on an event of probability at least $3/8$, and hence $\mathbb{E}[L_2] = \Omega_d(n^{\beta})$. In both cases $\mathbb{E}[\mathrm{Move}] = \Omega_d(n^{\beta})$. □

The next bound holds for every number of rounds and needs no schedule. It uses the cells of $\mathcal{C}_2$, which join the positions of ranks $2i-1$ and $2i$.

Proposition 4.5. *For every fixed d and every r, including $r = \infty$, $M_r^{(d)}(n) = \Omega_d\big(n^{1+1/(2d)}\big)$.*

Proof. Let r be finite. By Yao's principle, fix a deterministic r-round algorithm that is correct on every input and let π be uniform. Every cell $\{u, v\} \in \mathcal{C}_2$ joins the positions of ranks $2i-1$ and $2i$ for some i, and every correct algorithm must compare it. Indeed, if the algorithm never compares u and v, then exchanging their two ranks changes the outcome of no comparison that it makes. So, by induction over the rounds, it runs the same routes on both inputs and gives the same output, which is wrong on one of them. By Lemma 3.3 at scale 2, with probability at least $7/8$ at least half of the $\lfloor n/2 \rfloor$ cells $w \in \mathcal{C}_2$ satisfy $\mathrm{dist}(w, \mathcal{C}_2 \setminus \{w\}) > \rho$, where $\rho = \Theta_d(n^{1/(2d)})$, and these cells are pairwise more than ρ apart. The routes of all rounds, joined end to end, form one route of movement Move that visits all of them, so by Observation 2.3(b) and Fact 3.2, $\mathrm{Move} \ge \frac{1}{2} \cdot \frac{1}{2}\lfloor n/2 \rfloor \cdot \rho = \Omega_d(n^{1+1/(2d)})$ on this event. Hence $\mathbb{E}[\mathrm{Move}] = \Omega_d(n^{1+1/(2d)})$. The constant depends only on d, so the bound also holds for $M_\infty^{(d)}(n) = \inf_r M_r^{(d)}(n)$. □

4.3 Adaptive Rounds

Throughout Sections 4.3 and 4.4, let $r \ge 3$, fix a deterministic r-round algorithm in the augmented model that is correct on every input, let π be uniform, and write L_t for the movement of its route in round t. Since rounds $2, \dots, r-1$ are adaptive, we keep track of what the first $r-1$ rounds can have learned. The bad set contains, scale by scale, the positions that these rounds compare inside their blocks. It also contains all positions of every block in which more than half of the positions are already bad, so that every other block keeps at least half of its positions outside the bad set, which Lemma 4.8 uses.

Definition 4.6 (The Bad Set)**.** The *bad set* is built in levels $j = 0, 1, \dots, r-2$, and level j concerns round $j+1$. It starts from the set $\mathcal{A}_{-1}$ of positions whose rank lies in a partial block of some scale b_i. Level j has two steps.

(a) For $j \geq 1$, a scale-b_{j-1} block is *heavy at level* j if more than half of its positions lie in $\mathcal{A}_{j-1}$. Let $\mathcal{A}_j^-$ consist of $\mathcal{A}_{j-1}$ and all positions of the blocks that are heavy at level j. Let $\mathcal{A}_0^- = \mathcal{A}_{-1}$.

(b) Let X_j be the set of positions outside $\mathcal{A}_j^-$ that lie in a cell visited in round $j+1$ and co-blocked at scale b_j, and let $\mathcal{A}_j = \mathcal{A}_j^- \cup X_j$.

The bad set is $\mathcal{A} = \mathcal{A}_{r-2}$. A position is *bad* if it lies in $\mathcal{A}$, and *clean* otherwise. It is *clean at level* j if it lies outside $\mathcal{A}_j^-$.

Observation 4.7. (a) *For $1 \leq j \leq r-2$, a position clean at level j lies in a full scale-b_{j-1} block that is not heavy at level j, and in rounds $1, \ldots, j$ it has not been compared with another position of this block.*

(b) *A clean position has not been compared, in rounds $1, \ldots, r-1$, with another position of its scale-g block.*

Proof. For every i, both positions of a cell that round $i+1$ visits and that is co-blocked at scale b_i lie in $\mathcal{A}_i$. Let x be clean at level j, and suppose that in round $i+1 \leq j$ it was compared with another position y of its scale-b_{j-1} block. The cell $\{x, y\}$ is co-blocked at scale b_{j-1}, hence at the coarser scale b_i, so $x \in \mathcal{A}_i \subseteq \mathcal{A}_j^-$, a contradiction. The block of x is full because $\mathcal{A}_{-1} \subseteq \mathcal{A}_j^-$, and it is not heavy because otherwise $x \in \mathcal{A}_j^-$. This proves (a). For (b), let x be clean, and suppose that in round $i+1 \leq r-1$ it was compared with another position y of its scale-g block. The cell $\{x, y\}$ is co-blocked at scale g, hence at scale b_i, so $x \in \mathcal{A}_i \subseteq \mathcal{A}$, a contradiction. □

To keep the bad set small, we bound $\mathbb{E}|X_j|$ for $1 \leq j \leq r-2$ by combining two facts. A position enters X_j only if it is clean at level j and lies in a cell that round $j+1$ visits and that is co-blocked at scale b_j, and hence also at scale b_{j-1}. Lemma 4.8 below shows that a cell that is co-blocked at scale b_{j-1} and contains a position clean at level j is co-blocked at scale b_j with conditional probability $O(b_j/b_{j-1})$, given what the algorithm knows before round $j+1$, and Lemma 3.4 shows that a short route visits few cells co-blocked at scale b_{j-1}. Lemma 4.9 combines the two.

For $1 \leq j \leq r-2$, let $\mathcal{F}_j$ be the σ-field generated by the labels at scales $b_0, \ldots, b_{j-1}$ and the outcomes of rounds $1, \ldots, j$. In words, $\mathcal{F}_j$ is what the algorithm knows when it commits the route of round $j+1$. An *atom* of $\mathcal{F}_j$ is one possible value of these labels and outcomes. Given an atom, that route is fixed. The lemma below conditions on $\mathcal{F}_j$ and refers to the positions clean at level j, so we check that these positions are determined by $\mathcal{F}_j$. By induction on i, the sets $\mathcal{A}_{i-1}$ and $\mathcal{A}_i^-$ are determined by $\mathcal{F}_j$ for every $i \leq j$. The set $\mathcal{A}_{-1}$ is determined by the labels at scale b_0. Whether a block is heavy at level i depends only on $\mathcal{A}_{i-1}$ and the labels at scale b_{i-1}. For $i < j$, the set X_i depends only on $\mathcal{A}_i^-$, the route of round $i+1$, and the labels at scale b_i, and the last two are determined by $\mathcal{F}_{i+1} \subseteq \mathcal{F}_j$.

Lemma 4.8. *Let $1 \leq j \leq r-2$, and let $\{u, v\}$ be a cell. On the event that $\{u, v\}$ is co-blocked at scale b_{j-1} and u is clean at level j,*

$$\Pr\big[\{u, v\} \text{ is co-blocked at scale } b_j \;\big|\; \mathcal{F}_j\big] \leq \frac{4b_j}{b_{j-1}}.$$

Proof. Fix an atom of $\mathcal{F}_j$ on which $\{u, v\}$ is co-blocked at scale b_{j-1} and u is clean at level j, and let B be the scale-b_{j-1} block of u. By Observation 4.7(a), B is full and not heavy at level j. Since $\mathcal{A}_j^-$ adds to $\mathcal{A}_{j-1}$ only positions of heavy blocks, at most half of the b_{j-1} positions of B lie in $\mathcal{A}_j^-$, so the set Y_B of positions of B that are clean at level j has at least $b_{j-1}/2$ elements. Condition further

on the ranks of all positions outside Y_B. The ranks of B not assigned to these positions are the *free slots*. The positions of Y_B occupy them.

We show that rk restricted to Y_B is a uniformly random bijection onto the free slots. Since π is uniform, it suffices to show that permuting the ranks of Y_B among the free slots does not change the atom of $\mathcal{F}_j$. Such a permutation keeps every rank inside B, so it changes no label at scales $b_0, \dots, b_{j-1}$. By Observation 4.7(a), every comparison of rounds $1, \dots, j$ that involves a position of Y_B has its other position outside B. The rank of that position lies above all ranks of B or below all of them, so the outcome does not change. The outcome of a comparison that involves no position of Y_B does not change either. By induction over the rounds, the routes and outcomes of rounds $1, \dots, j$ do not change, so the atom of $\mathcal{F}_j$ does not change.

It remains to count. Since $\{u, v\}$ is co-blocked at scale b_{j-1}, v lies in B. If $v \notin Y_B$, the rank of v is fixed, and $\mathrm{rk}(u)$ is uniform over at least $b_{j-1}/2$ free slots, of which at most b_j lie in the scale-b_j block of $\mathrm{rk}(v)$, so the probability is at most $2b_j/b_{j-1}$. If $v \in Y_B$, condition also on $\mathrm{rk}(v)$. Then $\mathrm{rk}(u)$ is uniform over the remaining at least $b_{j-1}/2 - 1$ free slots, of which at most b_j lie in the scale-b_j block of $\mathrm{rk}(v)$, so the probability is at most $b_j/(b_{j-1}/2 - 1) \le 4b_j/b_{j-1}$, because $b_{j-1} \ge 2b_j \ge 4$. These bounds hold for every value of the further conditioning and every such atom of $\mathcal{F}_j$, which proves the lemma. □

Lemma 4.9. *We have* $\mathbb{E}|X_0| \le 2(L_1 + 1)\, b_0/n$, *and for every* $1 \le j \le r - 2$,

$$\mathbb{E}|X_j| = O_{d,r}\Big(\frac{b_j}{b_{j-1}}\Big(1 + \mathbb{E}[L_{j+1}]\Big(\frac{b_{j-1}}{n}\Big)^{1/D}\Big)\log n\Big).$$

Proof. No outcome and no label is known before round 1, so its route and L_1 do not depend on π, and by Observation 2.3(a) the route visits at most $L_1 + 1$ cells. Each of them is co-blocked at scale b_0 with probability at most b_0/n and contributes at most two positions to X_0, which gives the bound on $\mathbb{E}|X_0|$.

Now let $1 \le j \le r - 2$, and fix an atom of $\mathcal{F}_j$. The route of round $j + 1$ is then fixed, and so is the set K_j of the cells that it visits, that are co-blocked at scale b_{j-1}, and that contain a position clean at level j. Every position of X_j is clean at level j and lies in a cell visited in round $j + 1$ and co-blocked at scale b_j, hence also at scale b_{j-1}, so it lies in a cell of K_j that is co-blocked at scale b_j. Applying Lemma 4.8 to each cell of K_j, with u a position clean at level j, gives

$$\mathbb{E}\big[|X_j| \;\big|\; \mathcal{F}_j\big] \le 2 \cdot |K_j| \cdot \frac{4b_j}{b_{j-1}}.$$

We bound $|K_j|$ by Lemma 3.4 with $\alpha = a_0$, which applies at scale b_{j-1} because $a_0 < 1$ and $b_{j-1} \le b_0 \le n^{a_0}$. Its constant c then depends only on d and r. The event $E_{b_{j-1}}$ depends only on $\mathcal{C}_{b_{j-1}}$, which the labels at scale b_{j-1} determine, so it is determined by $\mathcal{F}_j$. On $E_{b_{j-1}}$ we have $|K_j| \le c\big(1 + L_{j+1}(b_{j-1}/n)^{1/D}\big)\log n$, so taking expectations gives

$$\mathbb{E}\big[|X_j|\,\mathbf{1}_{E_{b_{j-1}}}\big] \le \frac{8c\,b_j}{b_{j-1}}\Big(1 + \mathbb{E}[L_{j+1}]\Big(\frac{b_{j-1}}{n}\Big)^{1/D}\Big)\log n.$$

Off $E_{b_{j-1}}$ we use $|X_j| \le n$. The complement of $E_{b_{j-1}}$ has probability at most n^{-10} for large n, so it contributes at most n^{-9}, which is at most b_j/b_{j-1} because $b_j \ge 1$ and $b_{j-1} \le n$. Adding the two parts proves the bound on $\mathbb{E}|X_j|$. □

The first stage of the proof of Theorem 4.1 uses Lemma 4.9 together with the following observation.

Observation 4.10. *For* $0 \le j \le r-2$, $|\mathcal{A}_j| \le 3|\mathcal{A}_{j-1}| + |X_j|$. *Consequently* $\mathbb{E}|\mathcal{A}| \le 3^{r-1}\big(|\mathcal{A}_{-1}| + \max_{0\le j\le r-2} \mathbb{E}|X_j|\big)$.

Proof. At level $j \ge 1$, a full heavy block holds more than $b_{j-1}/2$ positions of $\mathcal{A}_{j-1}$, so at most $2|\mathcal{A}_{j-1}|/b_{j-1}$ full blocks are heavy, and together they hold at most $2|\mathcal{A}_{j-1}|$ positions. The positions of a partial block lie in $\mathcal{A}_{-1} \subseteq \mathcal{A}_{j-1}$ already. Hence $|\mathcal{A}_j^-| \le 3|\mathcal{A}_{j-1}|$, also for $j = 0$, and the first claim follows from $\mathcal{A}_j = \mathcal{A}_j^- \cup X_j$. Iterating it over the $r-1$ levels gives $|\mathcal{A}| \le 3^{r-1}|\mathcal{A}_{-1}| + \sum_{j=0}^{r-2} 3^{r-2-j}|X_j|$, and the second claim follows by taking expectations, because $\sum_{j=0}^{r-2} 3^{r-2-j} < 3^{r-1}$. □

Proof of Theorem 4.1. For $r \le 2$ the theorem is Proposition 4.4, so let $r \ge 3$. By Yao's principle, and because an algorithm of the original model is an algorithm of the augmented model that ignores the labels, it suffices to show that the algorithm fixed in Section 4.3 has $\mathbb{E}[\mathrm{Move}] = \Omega_{d,r}(n^\beta/\log n)$. The proof has the four stages of the proof of Proposition 4.4. The new difficulty is that rounds $2, \dots, r-1$ are adaptive. We first show that the bad set is small, then find many cells of $\mathcal{C}_g$ with clean positions that are pairwise far apart, show that round r visits all of them, and bound the movement of a route that does so.

Recall that L_t is the movement of the route of round t, so that $\mathrm{Move} = L_1 + \cdots + L_r$. Let $T = \varepsilon n^\beta/\log n$, where $\varepsilon > 0$ depends only on d and r and is chosen below. If $\mathbb{E}[L_t] \ge T/100$ for some $t \le r-1$, then $\mathbb{E}[\mathrm{Move}] \ge T/100$, and we call this Case 1. Otherwise $\mathbb{E}[L_t] < T/100$ for every $t \le r-1$. This is Case 2, which we assume until the last paragraph.

Exposure. We show that $\mathbb{E}|\mathcal{A}| \le n/64$ for large n. By Lemma 4.9 and Case 2, $\mathbb{E}|X_0| \le 2(T/100+1)b_0/n$. Since $b_0 \le n^{a_0}$ and, by Equation (2.1), $\beta + a_0 - 1 = 1$, this gives $\mathbb{E}|X_0| \le \varepsilon n/50 + 2$. For $1 \le j \le r-2$, let C be the constant of Lemma 4.9, which depends only on d and r. With Case 2 and $T\log n = \varepsilon n^\beta$, that lemma gives $\mathbb{E}|X_j| \le C\log n + C\varepsilon\,(b_j/b_{j-1})\,n^\beta (b_{j-1}/n)^{1/D}$. Since $b_j \le n^{a_j}$, $b_{j-1} > n^{a_{j-1}}/2$, and $1 - 1/D > 0$,

$$\frac{b_j}{b_{j-1}}\, n^\beta \Big(\frac{b_{j-1}}{n}\Big)^{1/D} = n^{\beta-1/D}\, b_j\, b_{j-1}^{-(1-1/D)} \le 2\, n^{\beta + a_j - a_{j-1} + (a_{j-1}-1)/D} = 2n,$$

where the last step is Equation (2.2). Hence $\mathbb{E}|X_j| \le C\log n + 2C\varepsilon n$. By Observation 4.10 and $|\mathcal{A}_{-1}| < b_0 \le n^{a_0}$, $\mathbb{E}|\mathcal{A}| \le 3^{r-1}\big(n^{a_0} + (2C+1/50)\,\varepsilon n + C\log n + 2\big)$. Choose ε with $3^{r-1}(2C+1/50)\,\varepsilon \le 1/128$. Then $\mathbb{E}|\mathcal{A}| \le n/64$ for large n, since $a_0 < 1$.

Survivors. Every position lies in fewer than g cells of $\mathcal{C}_g$, so at most $g|\mathcal{A}|$ cells of $\mathcal{C}_g$ have a bad position. Since $\mathbb{E}[g|\mathcal{A}|] \le ng/64 \le |\mathcal{C}_g|/8$, Markov's inequality applied to the number of these cells shows that, with probability at least $1/2$, at most $|\mathcal{C}_g|/4$ cells of $\mathcal{C}_g$ have a bad position. By Lemma 3.3 at scale g, with probability at least $7/8$ at least half of the cells $w \in \mathcal{C}_g$ satisfy $\mathrm{dist}(w, \mathcal{C}_g \setminus \{w\}) > \rho$, where $\rho = \Theta_d\big((n/g)^{1/D}\big)$. On the intersection of the two events, which has probability at least $1 - \frac12 - \frac18 = \frac38$, let $\mathcal{H}$ be the set of cells of $\mathcal{C}_g$ that have two clean positions and satisfy this condition. Then $|\mathcal{H}| \ge |\mathcal{C}_g|/4 \ge ng/32$, and any two cells of $\mathcal{H}$ are more than ρ apart.

Forcing. Let $\{u, v\} \in \mathcal{H}$. Both u and v are clean, so by Observation 4.7(b) neither has been compared, in rounds $1, \dots, r-1$, with another position of its scale-g block. By Lemma 4.3, applied to P_{r-1}, the positions u and v are incomparable in P_{r-1}. The revealed scales $b_0, \dots, b_{r-3}$ are coarser than g, and u and v are co-blocked at scale g, so they have the same label at every revealed scale. By Lemma 4.2, the route of round r visits $\{u, v\}$.

Cost of the Final Round. By the previous stage, the route of round r visits every cell of $\mathcal{H}$. Since $|\mathcal{H}| \geq 2$ and the cells of $\mathcal{H}$ are pairwise more than ρ apart, Observation 2.3(b) and Fact 3.2 give

$$L_r \;\geq\; \tfrac{1}{2}|\mathcal{H}|\rho \;=\; \Omega_d\big(ng\,(n/g)^{1/D}\big) \;=\; \Omega_d\big(n^{1+a_{r-2}+(1-a_{r-2})/D}\big) \;=\; \Omega_d(n^{\beta}),$$

where the second step uses $g > n^{a_{r-2}}/2$ and the last is Equation (2.3). So $L_r = \Omega_d(n^{\beta})$ on an event of probability at least $3/8$.

In Case 2 this gives $\mathbb{E}[L_r] = \Omega_d(n^{\beta})$, and in Case 1 we have $\mathbb{E}[\text{Move}] \geq T/100 = \varepsilon n^{\beta}/(100 \log n)$. In both cases $\mathbb{E}[\text{Move}] = \Omega_{d,r}(n^{\beta}/\log n)$, which proves Theorem 4.1. □

Remark 4.11. Theorem 4.1 holds with free computation and memory: comparisons are the algorithm's only access to the order. After Yao's principle fixes a deterministic algorithm, the proof of Lemma 4.2 gives, for every pair of positions with the same revealed labels that the first $r-1$ rounds leave incomparable, two inputs that agree on all revealed labels and on all outcomes of these rounds but order the pair oppositely. Hence no computation on these outcomes orders the pair, and the comparisons of the final round, with their movement, are required of every correct algorithm.

4.4 Exact Order on the Line

The logarithmic factor in Theorem 4.1 comes from Lemma 3.4 alone, and on the line we remove it. In this subsection $d = 1$, and we write $\beta_r = \beta_r^{(1)}$.

Theorem 4.12. *For $d = 1$ and every fixed $r \geq 1$, $M_r^{(1)}(n) = \Omega_r\big(n^{\beta_r}\big)$.*

Together with Theorem 5.11, it gives $M_r^{(1)}(n) = \Theta_r\big(n^{\beta_r}\big)$.

The logarithm in Lemma 3.4 comes from a product box that holds about $\log n$ cells, and such boxes are rare. So we group nearby cells into clusters. Small clusters are far apart for their size, so a route must move to visit their cells, and large clusters almost never occur. The bound that replaces Lemma 3.4 has an additive term Ξ_b, which counts the cells of large clusters and which we define first.

Fix a dyadic scale $b \leq \sqrt{n}$, and identify each cell with its fold, a point of $\{(x, y) : x < y\}$. By Observation 2.3(b), the distance dist between two cells is the ℓ_1 distance between their folds. Let $H = \lceil \sqrt{n/b} \rceil$, let $\varepsilon_\star > 0$ and n_0 be absolute constants chosen in the proof of Lemma 4.14, and put

$$\rho_t = \varepsilon_\star H\, 4^t \quad (0 \leq t \leq t_\star), \qquad t_\star = \lceil \log\log n \rceil, \qquad \kappa = 2^{t_\star}.$$

A *cluster* is a set of cells of $\mathcal{C}_b$. Clusters are formed in phases $t = 0, 1, \ldots, t_\star$, and the clusters of phase 0 are the single cells of $\mathcal{C}_b$. Two clusters of phase t are *close* if some cell of one lies within distance ρ_t of some cell of the other. Phase t merges each connected component of the closeness relation that contains two or more clusters into one cluster of phase $t+1$. A cluster of phase t that is close to no other cluster is *isolated*. An isolated cluster *retires* if it has at most 2^{t+1} cells. Otherwise it is also a cluster of phase $t+1$. The *residual clusters* are the clusters of phase $t_\star + 1$. For $n \geq n_0$, let Ξ_b be the total number of cells in residual clusters, and for $n < n_0$, let $\Xi_b = 0$.

By induction on t, every cluster of phase t has at least 2^t cells: a merged component consists of at least two clusters with at least 2^t cells each, and an isolated cluster that does not retire has more than 2^{t+1} cells. In particular every residual cluster has at least 2κ cells, and every cell of $\mathcal{C}_b$ ends in exactly one retired or residual cluster.

From one phase to the next, the distance ρ_t grows by a factor of 4, while the largest size of a retiring cluster only doubles. So the clusters that retire in a phase are far apart for their size.

Lemma 4.13. *Let $d = 1$. There are absolute constants C and C' such that, for every input and every dyadic scale $b \le \sqrt{n}$, every route of movement L visits at most $C\,L\sqrt{b/n} + \Xi_b + C' \log n$ cells of $\mathcal{C}_b$.*

Lemma 4.13 is the analog of Lemma 3.4 on the line, and the next lemma shows that its additive term Ξ_b is at most 1 in expectation.

Lemma 4.14. *Let $d = 1$. For a suitable choice of the absolute constants $\varepsilon_\star > 0$ and n_0, $\mathbb{E}[\Xi_b] \le 1$ over the uniform input, for every dyadic scale $b \le \sqrt{n}$.*

With these two lemmas, Lemma 4.9 has the following analog on the line.

Lemma 4.15. *We have $\mathbb{E}|X_0| \le 2(L_1 + 1)\, b_0/n$, and for every $1 \le j \le r-2$,*

$$\mathbb{E}|X_j| \le \frac{8b_j}{b_{j-1}}\Big(C\,\mathbb{E}[L_{j+1}]\Big(\frac{b_{j-1}}{n}\Big)^{1/2} + C' \log n + 1\Big),$$

where C and C' are the constants of Lemma 4.13.

Proof. The bound on $\mathbb{E}|X_0|$ is that of Lemma 4.9. Let $1 \le j \le r-2$. At $q = 1/2$ we have $\delta = 1/\big(2(1-2^{-r})\big) > 1/2$, so $a_0 = 1 - \delta < 1/2$ and $b_{j-1} \le b_0 < \sqrt{n}$, so Lemma 4.13 applies at scale b_{j-1}. We follow the proof of Lemma 4.9 with Lemma 4.13 in place of Lemma 3.4. As there, $\mathbb{E}\big[|X_j| \;\big|\; \mathcal{F}_j\big] \le (8b_j/b_{j-1})|K_j|$. By Lemma 4.13, $|K_j| \le C\,L_{j+1}(b_{j-1}/n)^{1/2} + \Xi_{b_{j-1}} + C' \log n$ on every input. Taking expectations and using $\mathbb{E}[\Xi_{b_{j-1}}] \le 1$ (Lemma 4.14) gives the bound. □

Proof of Theorem 4.12. For $r \le 2$ the theorem is Proposition 4.4, so let $r \ge 3$. As in the proof of Theorem 4.1, it suffices to show that the algorithm fixed in Section 4.3 has $\mathbb{E}[\mathrm{Move}] = \Omega_r(n^{\beta_r})$. We follow the proof of Theorem 4.1 with three changes.

First, $T = \varepsilon_r n^{\beta_r}$, where $\varepsilon_r > 0$ depends only on r and is chosen below. Cases 1 and 2 are defined with this T.

Second, the first stage uses Lemma 4.15 in place of Lemma 4.9. In Case 2 it gives $\mathbb{E}|X_0| \le 2(T/100 + 1)b_0/n \le \varepsilon_r n/50 + 2$, as in the proof of Theorem 4.1, and, for $1 \le j \le r-2$,

$$\mathbb{E}|X_j| \le \frac{8b_j}{b_{j-1}}\Big(\frac{C\,T}{100}\Big(\frac{b_{j-1}}{n}\Big)^{1/2} + C' \log n + 1\Big) \le \frac{C}{4}\,\varepsilon_r n + 8C' \log n + 8,$$

because $(b_j/b_{j-1})\, n^{\beta_r}(b_{j-1}/n)^{1/2} \le 2n$ by the computation in the proof of Theorem 4.1 with $D = 2$. By Observation 4.10, $\mathbb{E}|\mathcal{A}| \le 3^{r-1}\big(b_0 + (C/4 + 1/50)\,\varepsilon_r n + 8C' \log n + 10\big)$.

Third, choose ε_r with $3^{r-1}(C/4 + 1/50)\,\varepsilon_r \le 1/128$. Then $\mathbb{E}|\mathcal{A}| \le n/64$ for large n. The remaining stages use only this bound, and they give $\mathbb{E}[L_r] = \Omega_r(n^{\beta_r})$ in Case 2. In Case 1, $\mathbb{E}[\mathrm{Move}] \ge \varepsilon_r n^{\beta_r}/100$. Hence $\mathbb{E}[\mathrm{Move}] = \Omega_r(n^{\beta_r})$, which proves Theorem 4.12. □

Proof of Lemma 4.13. If $n < n_0$, then $\Xi_b = 0$, and a route visits at most $|\mathcal{C}_b| \le nb/2 \le n^{3/2} \le n_0^{3/2}$ cells of $\mathcal{C}_b$. Let $n \ge n_0$. Deleting the configurations (x, x) from a route of movement L and folding the others gives a sequence of folds of ℓ_1-length at most L (Observation 2.3(b)), and the cells of $\mathcal{C}_b$ that the route visits are those whose folds occur in this sequence. The residual clusters contain Ξ_b cells in all, and every other cell of $\mathcal{C}_b$ lies in a retired cluster. Fix a phase t. Two clusters that retire in phase t are isolated in phase t, so every cell of one is more than ρ_t from every cell of the other. Order the clusters retired in phase t that have a cell in the sequence by the first occurrence of one

of their cells. Consecutive first occurrences are more than ρ_t apart, so at most $1 + L/\rho_t$ of these clusters have a cell in the sequence, and each of them has at most 2^{t+1} cells. Summing over the phases, the route visits at most

$$\sum_{t=0}^{t_\star} 2^{t+1}\Big(1 + \frac{L}{\rho_t}\Big) \le 2^{t_\star+2} + \frac{L}{\rho_0}\sum_{t\ge 0}\frac{2^{t+1}}{4^t} = 2^{t_\star+2} + \frac{4L}{\rho_0}$$

cells of retired clusters. Since $2^{t_\star} < 2\log n$ and $\rho_0 = \varepsilon_\star H \ge \varepsilon_\star\sqrt{n/b}$, this is at most $8\log n + (4/\varepsilon_\star)L\sqrt{b/n}$. Adding the Ξ_b cells of residual clusters gives the bound for $n \ge n_0$. Hence the lemma holds with $C = 4/\varepsilon_\star$ and $C' = \max\{8, n_0^{3/2}\}$. □

Proof of Lemma 4.14. If $n < n_0$, then $\Xi_b = 0$, so let $n \ge n_0$, where n_0 is chosen at the end of the proof. We charge Ξ_b to sets of between κ and 2κ cells of $\mathcal{C}_b$ that carry a tree of short edges, bound the probability that a fixed set of cells lies in $\mathcal{C}_b$, and bound the expected number of such sets.

For a finite set W of cells, its *position graph* has the positions occurring in W as vertices and the cells of W as edges. Let $c(W)$ be its number of vertices minus its number of components, which is the number of rank constraints that $W \subseteq \mathcal{C}_b$ imposes, and let $h(W) = |W| - c(W)$ be its cyclomatic number. We say that W is *eligible* if some tree with vertex set W carries a phase $\ell(e) \in \{0, \dots, t_\star\}$ on each of its edges $e = \{w, w'\}$ such that $\|w - w'\|_1 \le \rho_{\ell(e)}$ for every edge and $\sum_e \ell(e) \le C_0|W|$, where $C_0 = 16$. A *witness* is an eligible subset of $\mathcal{C}_b$ with between κ and 2κ cells. Let N_s be the number of eligible subsets of $\mathcal{C}_b$ with s cells.

Charging. Let F be a residual cluster. Its merge history gives a tree with vertex set F: when phase t merges a component of close clusters, choose a spanning tree of the component and, for each of its edges, a pair of cells, one in each of the two clusters, at distance at most ρ_t, and these pairs become tree edges with phase t. A merge in phase t adds fewer edges than it merges clusters, and every cluster of phase t has at least 2^t cells, so at most $2^{-t}|F|$ tree edges have phase t, and the phases sum to at most $\sum_{t\ge 0} t\, 2^{-t}|F| = 2|F|$.

We cut this tree into connected pieces. Root the tree and collect edges from the leaves upward: at each vertex, add the edge to each child together with the edges left over at that child, and emit the collected edges as one piece as soon as they number at least $\kappa - 1$. Every collected set is connected, and the edges left over at a vertex number fewer than $\kappa - 1$, so each piece has between $\kappa - 1$ and $2\kappa - 3$ edges and hence between κ and 2κ cells. The pieces have disjoint edge sets, and fewer than $\kappa - 1$ edges remain at the root. The tree has $|F| - 1$ edges and $|F| \ge 2\kappa$, so there are more than $(|F| - \kappa)/(2\kappa) \ge |F|/(4\kappa)$ pieces. Their phase sums total at most $2|F|$, so fewer than $|F|/(8\kappa)$ pieces have phase sum above 16κ. Each of the remaining, more than $|F|/(8\kappa)$, pieces has a cell set W with $|W| \ge \kappa$ and phase sum at most $16\kappa \le C_0|W|$, so W is a witness. Distinct pieces have distinct cell sets, since a connected subtree with cell set W contains every tree edge inside W, and distinct residual clusters are disjoint. Hence each residual cluster F yields more than $|F|/(8\kappa)$ witnesses, all distinct, and

$$\Xi_b \;\le\; 8\kappa \sum_{s=\kappa}^{2\kappa} N_s.$$

Witness Probability. Fix a set W of $s \le 2\kappa$ cells. Reveal the ranks of its positions one component of the position graph at a time, each component in an order in which every vertex x after the first has a neighbor x' earlier in the order. If $W \subseteq \mathcal{C}_b$, then every such x receives a rank in the scale-b block of x'. Given the earlier ranks, this block contains at most b of the at least $n - 2s \ge n/4$

unrevealed ranks, so this happens with probability at most $4b/n$. There are $c(W)$ such vertices, so $\Pr[W \subseteq \mathcal{C}_b] \le (4b/n)^{c(W)} = (4b/n)^{s-h(W)}$.

First Moment. Let $N_s^{(h)}$ be the number of eligible subsets of $\mathcal{C}_b$ with s cells and cyclomatic number h. We bound $\mathbb{E}[N_s^{(h)}]$ for $h \le \eta s/t_\star$ here, where η is chosen below, and treat larger h in the next paragraph. Given an eligible set W with s cells, fix a tree as in the definition of eligibility, root it, order the children of each vertex, and add the cells of W one at a time in the preorder of this ordered rooted tree, so that every cell comes after its parent. Adding a cell increases c of the set added so far by one, unless its two positions already lie in one component of the position graph of that set. We then say that the cell *closes a cycle*. Since $c(W) = s - h$, exactly h cells close a cycle. Hence W is encoded by the ordered rooted tree, at most 4^s choices; the phases of its edges, which sum to at most $C_0 s$, at most $2^{(C_0+1)s}$ choices; the set of cells that close a cycle, at most 2^s choices; and the placement of the cells in preorder. A cell that closes a cycle is one of at most $(2s)^2$ pairs of positions already present. The root is one of at most $n^2/2$ cells. Every other cell lies within distance ρ_ℓ of its parent, where ℓ is the phase of the tree edge to its parent, so it has at most $(2\rho_\ell + 1)^2 \le 9\rho_\ell^2$ placements, since $\rho_\ell \ge \varepsilon_\star H \ge \varepsilon_\star n^{1/4} \ge 1$ for $n \ge n_0$. The bound $\Pr[W \subseteq \mathcal{C}_b] \le (4b/n)^{s-h}$ gives a factor $4b/n$ for each of the $s - h$ cells that close no cycle. Since $H^2 \le 4n/b$, we have $9\rho_\ell^2 \cdot 4b/n \le 144\,\varepsilon_\star^2 16^\ell$, and the phases give $\prod_e 16^{\ell(e)} \le 16^{C_0 s}$. Collecting the factors,

$$\mathbb{E}\big[N_s^{(h)}\big] \le 2nb\,\varepsilon_\star^{-2}\left(C_1\varepsilon_\star^2\right)^s\left(4s^2/\varepsilon_\star^2\right)^h$$

for an absolute constant C_1. Let $\varepsilon_\star^2 = 1/(16C_1)$ and $\eta = \big(6 + 2\log(1/\varepsilon_\star)\big)^{-1}$. If $h \le \eta s/t_\star$, then $(4s^2/\varepsilon_\star^2)^h \le 2^s$. Indeed, $s \le 2\kappa < 4\log n$ gives $\log(4s^2) \le 6 + 2\log\log n \le 6t_\star$ for $n \ge n_0$, so $h\log(4s^2/\varepsilon_\star^2) \le (\eta s/t_\star)\big(6t_\star + 2\log(1/\varepsilon_\star)\big) \le s$. Hence $\mathbb{E}[N_s^{(h)}] \le 2nb\,\varepsilon_\star^{-2}8^{-s}$, and summing over the at most s values of h gives $\sum_{h \le \eta s/t_\star}\mathbb{E}[N_s^{(h)}] \le nb\,4^{-s}$ for $s \ge \kappa$ and $n \ge n_0$.

High Cyclomatic Number. Let N_s^{high} be the number of eligible subsets W of $\mathcal{C}_b$ with s cells and $h(W) > \eta s/t_\star$. For these we use the geometry of the set instead. Tree edges join cells at distance at most $\rho_{t_\star} = \varepsilon_\star H\kappa^2$, so every cell of W lies within distance $(s-1)\rho_{t_\star}$ of the root, and all positions of W lie in the two intervals of radius $R = s\rho_{t_\star}$ around the two positions of the root. These intervals contain $V \le 4R + 2 = O\big(\sqrt{n/b}\,\log^3 n\big)$ positions. Let $N_{s,\omega,\tau}$ be the number of these sets whose position graph has ω vertices and τ components. Such a set is encoded by its root, at most n^2 choices, its other $\omega - 2$ positions, at most $V^{\omega-2}$ choices, and its s cells as pairs of its positions, at most $(2s)^{2s}$ choices. With $\gamma = 4b/n$ and $z = \omega - 2\tau$, the bound on $\Pr[W \subseteq \mathcal{C}_b]$ gives

$$\mathbb{E}[N_{s,\omega,\tau}] \le n^2(2s)^{2s}V^{\omega-2}\gamma^{\omega-\tau} = n^2(2s)^{2s}V^{-2}\left(V^2\gamma\right)^\tau\left(V\gamma\right)^z.$$

Here $V^2\gamma = O(\log^6 n)$, and $V\gamma \le n^{-1/5}$ for $n \ge n_0$ because $b \le \sqrt{n}$. The position graph is simple, so a component with two vertices contributes nothing to z or to h. A component with $\omega' \ge 3$ vertices contributes $\omega' - 2 \ge \omega'/3$ to z, so $z \ge \omega_3/3$, where ω_3 is the number of vertices in components with at least three vertices. Also $h \le \omega_3^2/2$, because h is at most the number of edges in these components. Hence $z \ge \sqrt{2h}/3$, and $h > \eta s/t_\star \ge \eta\kappa/t_\star$ gives $z = \Omega\big(\sqrt{\log n/\log\log n}\big)$. So $(V\gamma)^z = n^{-\Omega(\sqrt{\log n/\log\log n})}$, whereas $n^2(2s)^{2s}(V^2\gamma)^\tau = \exp\big(O(\log n\,\log\log n)\big)$ because $s, \tau \le 4\log n$. For $n \ge n_0$ the product is at most n^{-22}, and summing over the at most $(2s)^2 \le n^2$ pairs (ω, τ) gives $\mathbb{E}[N_s^{\mathrm{high}}] \le n^{-20}$.

For $\kappa \le s \le 2\kappa$ the two regimes give $\mathbb{E}[N_s] \le nb\,4^{-s} + n^{-20}$. Since $4^{-\kappa} \le n^{-2}$ and $b \le \sqrt{n}$,

$$\mathbb{E}[\Xi_b] \;\le\; 8\kappa\sum_{s=\kappa}^{2\kappa}\mathbb{E}[N_s] \;\le\; 8\kappa(\kappa+1)\big(n^{-1/2} + n^{-20}\big) \;\le\; 1,$$

where n_0 is chosen so large that this inequality and the estimates above hold for all $n \geq n_0$. □

5 Routing and the Sampling Algorithm

The upper bounds of Theorem 1.1 are attained by algorithms in which every route visits configurations known when the round starts, and Section 5.1 bounds the movement this takes. A merge sort then gives the fully adaptive bound (Section 5.2). For a fixed number $r \geq 2$ of rounds, the sampling algorithm refines the rank order round by round (Section 5.3). On the line, each refinement round selects intervals of one length (Section 5.4), and Section 6 treats $d \geq 2$.

5.1 Routing

The next lemma bounds the movement needed to visit a known set of configurations.

Lemma 5.1. (i) *Let $\mathcal{Q}$ be a set of $Q \geq 1$ points of a box in $\mathbb{Z}^{D'}$ whose sides are at most m'. From any given point of the box, a lattice path inside the box, that is, a sequence of points with consecutive points at ℓ_1 distance 1, visits all points of $\mathcal{Q}$ and has length $O_{D'}\big(m'Q^{1-1/D'} + m'\big)$. The path can be computed from $\mathcal{Q}$ in polynomial time.*

(ii) *In the comparison grid, any $Q \geq 1$ configurations are visited by a route that starts at the current configuration and has movement $O_d\big(n^{1/d}Q^{1-1/D} + n^{1/d}\big)$.*

Proof. For (i), we cut the box into small cubes and traverse them in *serpentine order*, visiting the points of $\mathcal{Q}$ in each cube on the way. This order is lexicographic, except that the direction along each axis is reversed at every step along an earlier axis. Let $\Delta = \lceil m'Q^{-1/D'} \rceil$ and cut the box into cubes of side Δ. There are at most $(m'/\Delta + 1)^{D'} \leq (Q^{1/D'} + 1)^{D'} \leq 2^{D'}Q$ of them. Consecutive cubes of the serpentine order are adjacent, so the path moves $O_{D'}(\Delta)$ from one cube to the next. Inside a cube it moves $O_{D'}(\Delta)$ from one point of $\mathcal{Q}$ to the next. Together with the $O_{D'}(m')$ steps from the given point to the first cube, this gives length $O_{D'}(\Delta Q + m')$. Since the box has at most $m'^{D'}$ points, $Q^{1/D'} \leq m'$, so $\Delta Q \leq m'Q^{1-1/D'} + Q \leq 2m'Q^{1-1/D'}$.

Part (ii) is (i) for the box $[m]^D$, with $D' = D$ and $m' = m = n^{1/d}$. □

Merging two sorted lists by walking them in rank order can move the heads far for every comparison. The next lemma instead advances many binary searches in one round, so that one route serves them all.

Lemma 5.2. *Let $R \subseteq [m]^d$ be a box of aspect ratio $O_d(1)$ with $|R| \geq 2$, and let its positions be partitioned into two lists, each sorted by rank. A deterministic algorithm finds the rank order of R in $O\big((\log|R|)^2\big)$ rounds with total movement $O_d\big(|R|^{1+1/(2d)}\big)$. Its routes stay in $R \times R$, and the bound holds whichever configuration of $R \times R$ each round starts from.*

Proof. The algorithm splits the two lists at the median of their union by binary search, recurses on the two parts, and advances all binary searches of one depth together. Write $N = |R|$. A subproblem consists of a sublist of consecutive entries of each list, s entries in all, and asks for the $t = \lceil s/2 \rceil$ lowest keys of their union. For $1 \leq i \leq t$, the i-th entry of the first sublist, if it exists, has $i - 1$ entries of its sublist below it, so it is among these t keys exactly when at most $t - i$ entries of the second sublist lie below it. Equivalently, it lies below the $(t - i + 1)$-th entry of the second sublist, or that entry does not exist. This condition is monotone in i, so a binary search on i finds the t lowest keys with $O(\log(s + 2))$ comparisons of one entry of each sublist. The t lowest keys form a prefix of

each sublist, so the two parts are again subproblems, of sizes $\lceil s/2 \rceil$ and $\lfloor s/2 \rfloor$. The recursion stops at subproblems with an empty sublist, which are already sorted, and this gives the rank order of R.

Each round advances every binary search of the current depth by one comparison. At depth λ there are at most 2^λ subproblems of size at most $\lceil N/2^\lambda \rceil$, so this depth takes $O\big(\log(N/2^\lambda + 2)\big)$ rounds. There are $O(\log N)$ depths, which gives the bound on the rounds. A round at depth λ visits at most 2^λ known configurations of $R \times R$, a box whose sides are $O_d(N^{1/d})$. By Lemma 5.1(i), from any given configuration of $R \times R$ its movement is $O_d\big(N^{1/d} 2^{(1-1/D)\lambda} + N^{1/d}\big)$. Substituting $i = \lceil \log_2 N \rceil - \lambda$, the main terms total

$$O_d\Big(N^{1/d} \sum_\lambda 2^{(1-1/D)\lambda} \log\big(N/2^\lambda + 2\big)\Big) = O_d\Big(N^{1/d} N^{1-1/D} \sum_{i \geq 0} (i+2)\, 2^{-(1-1/D)i}\Big) = O_d\big(N^{1+1/(2d)}\big),$$

since the series converges and $1/d - 1/D = 1/(2d)$. The additive terms total $O_d\big(N^{1/d} (\log N)^2\big) = O_d\big(N^{1+1/(2d)}\big)$. □

5.2 The Fully Adaptive Bound

For $r = \infty$, a merge sort over a partition of the grid into boxes gives the upper bound in every dimension. Together with Proposition 4.5, it determines $M_\infty^{(d)}(n)$.

Proposition 5.3. *For every $d \geq 1$, a deterministic algorithm sorts in $O\big((\log n)^3\big)$ rounds with movement $O_d\big(n^{1+1/(2d)}\big)$. Hence $M_\infty^{(d)}(n) = O_d\big(n^{1+1/(2d)}\big)$.*

Proof. We sort by merging along a tree of boxes of depth $T = d\lceil \log_2 m \rceil \leq \log_2 n + d$. The root is $[m]^d$. At depth t, every box whose side along axis $t \bmod d$ is $s \geq 2$ is halved along that axis, into sides $\lceil s/2 \rceil$ and $\lfloor s/2 \rfloor$. A box whose side along this axis is 1 has itself as its only child. At depth T all boxes are single positions. The boxes of one depth form a grid and have aspect ratio $O(1)$. At height i, that is, at depth $T - i$, there are at most $2^{T-i} \leq 2^d n/2^i$ boxes, and each has at most 2^i positions, because a box has at most two children. The algorithm processes the heights in increasing order and merges the two halves of each halved box by Lemma 5.2.

By Lemma 5.2, the movement inside the boxes of height i is $O_d\big((n/2^i) \cdot 2^{i(1+1/(2d))}\big) = O_d\big(n\, 2^{i/(2d)}\big)$. Summing over $i \leq T$ gives a geometric series dominated by its last term, so the total is $O_d\big(n\, 2^{T/(2d)}\big) = O_d\big(n^{1+1/(2d)}\big)$.

The merges of one height share their rounds: a round of the algorithm is one round of each merge, and its route visits the boxes in the serpentine order of their grid. The heads cross from one box to the next with movement $O_d(2^{i/d})$, so these crossings, together with the move to the first box, add $O_d\big((n/2^i)\, 2^{i/d} + n^{1/d}\big) = O_d(n)$ per round. Each merge takes $O\big((\log n)^2\big)$ rounds and there are $O(\log n)$ heights, so the algorithm has $O\big((\log n)^3\big)$ rounds and the crossings total $O_d\big(n(\log n)^3\big) = O_d\big(n^{1+1/(2d)}\big)$. □

Remark 5.4. At $d = 1$ the tree of boxes is the merge tree of ordinary merge sort, with the merges of Lemma 5.2. The exponent $3/2$ agrees with the bound of Maass and Schnitger for sorting on a Turing machine with one worktape and a two-way read-only input tape [MS86; DMS91]. Their input consists of bit strings, so the agreement is up to logarithmic factors.

Algorithm 1 The sampling algorithm

Parameters: $r \geq 2$; the exponents $a_0 > \cdots > a_{r-2}$ of Lemma 2.5, with $b_j = n^{a_j}$; the samples of each refinement round; a route procedure for each refinement round and for the cleanup round.

Round 1 **(splitter).** Choose $\lceil n/b_0 \rceil$ positions uniformly at random without replacement, the splitters.
The required configurations are all (u, v) with $u \neq v$ and v a splitter. For each splitter v in turn, place the second head on v and sweep the first head through $[m]^d$.
The splitters become resolved keys. In rank order, they cut the other positions into sets, and the nonempty sets are the segments.
for $j = 1, \ldots, r-2$ **do**
    Round $j+1$ **(refinement).** Draw the samples of this round independently. The pivots of the round are the pivots of its samples.
    The required configurations are all (u, v) with $u \neq v$ and v a pivot in the segment of u. Visit them by the route procedure.
    The pivots become resolved keys. In each segment, the nonempty maximal pivot-free sets of consecutive ranks become the new segments.
end for
Round r **(cleanup).** The required configurations are, for each cell whose two positions lie in one segment, one fixed orientation. Visit them by the route procedure.
Sort each segment, and output the resolved keys and the segments in rank order.

5.3 The Sampling Algorithm

Between rounds, the sampling algorithm knows the rank order only up to a partition of the keys into sets of consecutive ranks.

Definition 5.5 (Segment)**.** A *segment* is a nonempty set of positions whose ranks form an interval of $[n]$ known to the algorithm. A segment may be a single position.

The comparisons that remain are the pairs within segments, and the movement they take depends on where each segment lies: a segment with many keys in a small region needs little movement. The following quantity records this, by weighting each key with the number of boxes that its segment meets.

Definition 5.6 (Dispersion)**.** For a partition $\mathcal{P}$ of $[m]^d$ into boxes and a family of disjoint segments, let $\mathcal{P}(G)$ be the set of boxes of $\mathcal{P}$ that a segment G meets. The *dispersion* of the family is

$$S_{\mathcal{P}} = \sum_{G} |G|\,|\mathcal{P}(G)|.$$

Splitting a segment never raises the dispersion: if $G_1, G_2 \subseteq G$ are disjoint, then $\mathcal{P}(G_i) \subseteq \mathcal{P}(G)$, so $|G_1|\,|\mathcal{P}(G_1)| + |G_2|\,|\mathcal{P}(G_2)| \leq |G|\,|\mathcal{P}(G)|$.

The algorithm has r rounds. The first chooses random positions, the *splitters*. The next $r-2$ are *refinement rounds*, which draw samples, and the last is the *cleanup*. A *sample* is given by a partition $\mathcal{P}$ of $[m]^d$ into boxes and a rate $p \in (0, 1]$. Drawing it selects each box of $\mathcal{P}$ independently with probability p, and the keys in selected boxes that lie in segments are its *pivots*. The splitters and the pivots are called *resolved keys*, since the order of each of them relative to every other key is known after its round. Resolved keys lie in no segment.

Algorithm 1 is a template: each instance below specifies the samples and the routes. The state between rounds is the sequence of resolved keys and nonempty segments in rank order, and the analysis tracks the dispersion of the segments. The required configurations of a round are known when the round starts, so its route depends only on the coins and on the outcomes of earlier rounds, and in every instance it is computed in polynomial time. Line 2 moves $O(n)$ per splitter, so round 1 has movement $O(n\lceil n/b_0\rceil) = O(n^{2-a_0})$, and $2 - a_0 = \beta_r^{(d)}$ by Equation (2.1).

The algorithm is correct for every input and every outcome of its coins. Every key other than a splitter is compared with all splitters in round 1, so its segment is known. In a refinement round, every key of a segment, pivots included, is compared with all other pivots of this segment, so the new state is known. The cleanup compares all pairs within a segment, and the state determines the order of every other pair.

The next lemma bounds the *squared mass* $A = \sum_G |G|^2$ of the family after round 1, which counts the ordered pairs of keys in a common segment. The squared mass is the dispersion for the partition into single positions, where $|\mathcal{P}(G)| = |G|$.

Lemma 5.7. *Fix an input and choose s positions uniformly at random without replacement, where $1 \le s \le n$. In rank order, the chosen positions cut the other positions into sets $G_0, \dots, G_s$, which may be empty. Then $\mathbb{E}\sum_i |G_i|^2 \le 3n^2/s$.*

Proof. The sum $\sum_i |G_i|^2$ counts the ordered pairs of ranks that lie in a common set, and we bound the probability of each pair. Since the input is a bijection, the chosen ranks form a uniformly random set of s ranks. Two ranks at distance Δ lie in a common set only if none of the Δ ranks starting at the lower one is chosen, which has probability at most $(1 - s/n)^\Delta \le \exp(-s\Delta/n)$. Hence

$$\mathbb{E}\sum_i |G_i|^2 \;\le\; n + 2n\sum_{\Delta\ge 1}\exp(-s\Delta/n) \;\le\; n + \frac{2n^2}{s} \;\le\; \frac{3n^2}{s},$$

using $\sum_{\Delta\ge1}\exp(-s\Delta/n) = (\exp(s/n) - 1)^{-1} \le n/s$. □

A sample of rate p leaves dispersion about n/p for its own partition, in every dimension and whatever the family that enters the round.

Lemma 5.8. *Fix an input and a family of disjoint segments, and draw a sample with partition $\mathcal{P}$ and rate p. Within each segment, the pivots split the other keys into* refined segments*, the nonempty maximal pivot-free sets of consecutive ranks. For a key u in a segment, let G_u be its refined segment if u is not a pivot, and $G_u = \emptyset$ otherwise. Then $\mathbb{E}|\mathcal{P}(G_u)| \le 3/p$, and hence the dispersion of the refined segments satisfies $\mathbb{E}[S_\mathcal{P}] \le 3n/p$.*

Proof. We bound the number of boxes that G_u meets above and below u, and then sum over the keys. List the segment of u in rank order. Let R_u^+ be the number of boxes of $\mathcal{P}$, other than the box of u, that the keys above u meet before the first pivot above u. If $R_u^+ \ge t$, then the first t such boxes in rank order are all unselected. These boxes are determined by the input and the family, and each is selected independently with probability p, so $\Pr[R_u^+ \ge t] \le (1-p)^t$, and $\mathbb{E}[R_u^+] \le 1/p$. Define R_u^- in the same way for the keys below u. Then $|\mathcal{P}(G_u)| \le 1 + R_u^- + R_u^+$, whose expectation is at most $3/p$. Finally, $S_\mathcal{P} = \sum_u |\mathcal{P}(G_u)|$, and there are at most n keys. □

The later rounds tour the product boxes of a partition and visit the required configurations inside each of them. Fix a family of disjoint segments and a partition $\mathcal{P}$ of $[m]^d$ into boxes. For $I, J \in \mathcal{P}$,

Algorithm 2 The line algorithm

Parameters: $d = 1$; $r \geq 2$; h_j and p_j as above.

1: **Round** 1 **(splitter).** Lines 1–3 of Algorithm 1.
2: **for** $j = 1, \ldots, r-2$ **do**
3: **Round** $j+1$ **(refinement).** Draw one sample: intervals of length h_j, rate p_j.
4: **for** each selected interval J, in order on the line **do**
5: Place the second head at the left endpoint of J, and sweep the first head along the line from left to right.
6: When the first head reaches an interval I with $Q_{I,J} > 0$, visit the required configurations in $I \times J$ by Lemma 5.1(i). Then return the second head to the left endpoint of J and the first head to its place in the sweep.
7: **end for**
8: Update the state as in Line 7 of Algorithm 1.
9: **end for**
10: **Round** r **(cleanup).** With $h = h_{r-1}$, move along the lattice path of Lemma 5.1(i) through the occupied product boxes on the grid of $\lceil n/h \rceil^2$ product boxes. On first arrival at an occupied $I \times J$, visit the required configurations in it by Lemma 5.1(i).
11: Output as in Line 10 of Algorithm 1.

the set $I \times J$ is a product box, and we write $Q_{I,J} = \sum_G |G \cap I|\,|G \cap J|$. The product box $I \times J$ is *occupied* if $Q_{I,J} > 0$, and P is the number of occupied product boxes. If this family enters a refinement round or the cleanup, a required configuration in $I \times J$ is a pair (u, v) with $u \in I$, $v \in J$, and u and v in a common segment, so at most $Q_{I,J}$ required configurations lie in $I \times J$. The number of occupied product boxes bounds the tour, and on the line the sum of the $\sqrt{Q_{I,J}}$ bounds the visits. The dispersion bounds both:

$$P \;\leq\; \sum_G |\mathcal{P}(G)|^2 \;\leq\; S_{\mathcal{P}}, \qquad \sum_{I,J} \sqrt{Q_{I,J}} \;\leq\; \sum_G \Big(\sum_I \sqrt{|G \cap I|} \Big)^2 \;\leq\; S_{\mathcal{P}}. \tag{5.1}$$

For the first chain, a segment contributes to $Q_{I,J}$ only if it meets both I and J, which holds for $|\mathcal{P}(G)|^2$ product boxes, and $|\mathcal{P}(G)| \leq |G|$. For the second chain, the first step is the subadditivity of the square root over the segments, and the second is the Cauchy–Schwarz inequality over the $|\mathcal{P}(G)|$ nonzero terms $\sqrt{|G \cap I|}$.

5.4 The Upper Bound on the Line

On the line, the dispersion bounds both the tour of the next round and its visits (Equation (5.1)), so each refinement round draws one sample. In this subsection $d = 1$, so $D = 2$ and $q = \frac{1}{2}$, and we write $\beta = \beta_r$. An *interval of length* h is a set of h consecutive positions. We partition the line into intervals of length h, the last possibly shorter, and write $\mathcal{P}_h(G)$ and S_h for this partition. An interval of length h meets at most two intervals of length $h' \geq h$, so $S_{h'} \leq 2S_h$.

For $1 \leq j \leq r-1$, let $h_j = \lceil \sqrt{n/b_{j-1}} \rceil$, so that $h_1 \leq \cdots \leq h_{r-1} = \lceil \sqrt{n/g} \rceil$, and for $1 \leq j \leq r-2$, let $p_j = n^{-a_j}$. Algorithm 2 is the instance of Algorithm 1 with these parameters. We also apply Lemma 5.1(i) to coarser grids whose points are intervals, cubes, or product boxes. A step of the lattice path then moves the heads to an adjacent interval, cube, or product box.

The next two lemmas bound the movement of these routes for any family of segments.

Lemma 5.9. *Let a family of disjoint segments have dispersion S_h for intervals of length h, and let $p \in (0,1]$. For a sample with intervals of length h and rate p, the route of Lines 4–6 of Algorithm 2 visits all required configurations and has expected movement $O\big(p\,n^2/h + p\,h\,S_h + n\big)$.*

Proof. The route visits all required configurations, because every required configuration lies in a product box $I \times J$ with J selected and $Q_{I,J} > 0$. The first head moves at most $2n$ per selected interval on its sweep, and the second head moves at most $2n$ between selected intervals over the whole round. In expectation there are at most $p(n/h+1)$ selected intervals, so these moves total $O(p\,n^2/h + n)$. For an occupied $I \times J$ with J selected, at most $Q_{I,J}$ required configurations lie in $I \times J$, so by Lemma 5.1(i) visiting them and returning takes movement $O\big(h\sqrt{Q_{I,J}}\big)$, because $Q_{I,J} \geq 1$. Each J is selected with probability p, so by Equation (5.1) these visits have expected total $O\big(p\,h\sum_{I,J}\sqrt{Q_{I,J}}\big) = O(p\,h\,S_h)$. □

Lemma 5.10. *Let a family of disjoint segments have dispersion S_h for intervals of length h. The route of Line 10 of Algorithm 2, run with this h, visits the required configurations of the cleanup and has movement $O\big(n\sqrt{S_h} + h\,S_h + n\big)$.*

Proof. By Equation (5.1), $P \leq S_h$. By Lemma 5.1(i) on the grid of $\lceil n/h \rceil^2$ product boxes, the lattice path has $O\big(\lceil n/h \rceil\sqrt{P} + \lceil n/h \rceil\big)$ steps, each of movement $O(h)$, so it contributes $O\big(n\sqrt{S_h} + n\big)$. Each occupied $I \times J$ holds at most $Q_{I,J}$ required configurations, so by Lemma 5.1(i) visiting them and returning takes movement $O\big(h\sqrt{Q_{I,J}}\big)$, because $Q_{I,J} \geq 1$. By Equation (5.1), these visits total $O(h\,S_h)$. □

Theorem 5.11. *For every fixed r, an r-round algorithm sorts the line with expected movement $O_r\big(n^{\beta_r}\big)$ on every input, and its output is correct for every outcome of its coins. For $r \geq 2$ this is Algorithm 2.*

Proof. For $r = 1$, a serpentine route through $[n]^2$ visits every cell, with movement $O(n^2) = O(n^{\beta_1})$, and it is deterministic. Let $r \geq 2$. Correctness was shown after Algorithm 1. We bound the rounds in turn, writing L_t for the movement of round t. The expectations are over the coins.

Round 1 has movement $O(n^\beta)$, as noted after Algorithm 1. By Lemma 5.7 with $s = \lceil n/b_0 \rceil$, its output has squared mass A with $\mathbb{E}[A] \leq 3nb_0$.

We first bound the dispersion of the family that enters each later round: for $1 \leq j \leq r-1$, the family entering round $j+1$ satisfies $\mathbb{E}[S_{h_j}] \leq 6nb_{j-1}$. For $j = 1$ this holds because A is the dispersion for intervals of length 1, so $S_{h_1} \leq 2A$. For $j \geq 2$, Lemma 5.8 applied to round j gives $\mathbb{E}[S_{h_{j-1}}] \leq 3n/p_{j-1} = 3nb_{j-1}$, whatever the family entering round j, and $S_{h_j} \leq 2S_{h_{j-1}}$ because $h_{j-1} \leq h_j$.

Now let $1 \leq j \leq r-2$. By Lemma 5.9,

$$\mathbb{E}[L_{j+1}] = O\big(p_j n^2/h_j + p_j h_j\, nb_{j-1} + n\big) = O\big(p_j\, n\sqrt{nb_{j-1}} + n\big),$$

because $\sqrt{n/b_{j-1}} \leq h_j \leq 2\sqrt{n/b_{j-1}}$. The exponent of $p_j\, n\sqrt{nb_{j-1}}$ is $\frac{3}{2} + a_{j-1}/2 - a_j$, which equals β by Equation (2.2) with $D = 2$. Hence $\mathbb{E}[L_{j+1}] = O(n^\beta)$.

For round r, let $h = h_{r-1}$. By Lemma 5.10 and Jensen's inequality for the square root,

$$\mathbb{E}[L_r] = O\big(n\sqrt{ng} + \sqrt{n/g}\cdot ng + n\big) = O\big(n^{3/2}\sqrt{g}\big).$$

The exponent $\frac{3}{2} + a_{r-2}/2$ equals β by Equation (2.3) with $D = 2$. Summing over the r rounds gives $\mathbb{E}[\text{Move}] = O_r(n^\beta)$. □

Together with Theorem 4.12, this gives the exact order on the line.

Remark 5.12. On the line, $\beta_r = 1 + \frac{1}{2}s_r$, where $s_r = 1 + 1/(2^r - 1)$ is the exponent of the worst-case number of comparisons that a deterministic algorithm needs to find the maximum of n keys, or to merge two sorted lists, in r rounds [Val75; HH82]. Here $g = n^{1/(2^r-1)}$, so the finest scale of Section 4 has $\Theta(ng) = \Theta(n^{s_r})$ co-blocked cells. Unless an earlier round already has expected movement $\Omega(n^{\beta_r})$, with constant probability the final round must visit a constant fraction of them, pairwise at distance $\Omega\big((n/g)^{1/2}\big)$, which takes movement $\Omega(n^{\beta_r})$.

6 Higher Dimensions

For $d \geq 2$ the movement of a round depends on the dispersion of its input at more than one box size, and a refinement round controls each size with a sample of its own. The three ranges of r use three kinds of refinement: none for $r \leq 2$ (Section 6.1), two independent samples in one round for $r = 3$ (Section 6.2), and one sample for each box size of a nested partition for $r \geq 4$ (Section 6.3). The section ends with the proof of Theorem 1.1.

6.1 At Most Two Rounds

For $r \leq 2$ there is no refinement round: a serpentine route sorts in one round, and for $r = 2$ the cleanup after the splitters is a visit to a known set of configurations.

Proposition 6.1. *Let $d \geq 2$. Then (i) $M_1^{(d)}(n) = O_d(n^2)$, attained by a deterministic algorithm, and (ii) $M_2^{(d)}(n) = O_d\big(n^{\beta_2^{(d)}}\big)$, where $\beta_2^{(d)} = \frac{6d-1}{4d-1}$.*

Proof. For $r = 1$, a serpentine route through $[m]^D$ visits every cell, with movement $O_d(n^2)$, and it is deterministic.

For $r = 2$, we run Algorithm 1 without refinement rounds, with $b_0 = n^{a_0}$ for the exponent a_0 of Lemma 2.5 at $r = 2$. Round 1 has movement $O(n^\beta)$, where $\beta = \beta_2^{(d)}$. By Lemma 5.7 with $s = \lceil n/b_0 \rceil$, its output has squared mass A with $\mathbb{E}[A] \leq 3nb_0$. Round 2 has at most A required configurations, and they are known, so by Lemma 5.1(ii) and Jensen's inequality for the concave function $t \mapsto t^{1-1/D}$, its expected movement is $O_d\big(n^{1/d}(nb_0)^{1-1/D} + n^{1/d}\big)$. The exponent $\frac{1}{d} + (1 + a_0)\big(1 - \frac{1}{D}\big)$ equals $1 + a_0 + (1 - a_0)/D$, because $1/d = 2/D$, and this is β by Equation (2.3). □

6.2 Two Samples at $r = 3$

The cost of the cleanup has two parts. Its route moves along a lattice path through the occupied product boxes $I \times J$ of two cubes of side H, and inside each of them it visits the required configurations, with movement $O_d\big(H\, Q_{I,J}^{1-1/D} + H\big)$ by Lemma 5.1(i). On the line, $1 - 1/D = \frac{1}{2}$, so by Equation (5.1) the dispersion bounds both parts, and one sample of intervals suffices. For $d \geq 2$ the exponent $1 - 1/D$ is larger, and we bound a visit by $H\, Q_{I,J}$, whose sum over the boxes is H times the squared mass. Sampling cubes controls the dispersion, which bounds the lattice path, and sampling keys controls the squared mass, which bounds the visits, so the refinement round does both.

We write $\mathcal{P}_H(G)$ and S_H for the partition of $[m]^d$ into cubes of side H.

Algorithm 3 The two-sample algorithm

Parameters: $d \geq 2$; H, p, and σ as above.

1: **Round** 1 **(splitter).** Lines 1–3 of Algorithm 1.
2: **Round** 2 **(refinement).** Draw a cube sample of side H and rate p and a key sample of rate σ, independently.
3: **for** each selected cube J with $P_J \geq 1$, in the serpentine order of the cubes **do**
4: Place the second head at the minimum corner of J, and move the first head along the lattice path of Lemma 5.1(i) through the cubes I with $Q_{I,J} > 0$.
5: On arriving at such a cube I, visit the required configurations of the cube sample in $I \times J$ by Lemma 5.1(i). Then return the second head to the minimum corner of J and the first head to its place on the path.
6: **end for**
7: Visit the required configurations of the key sample by Lemma 5.1(ii).
8: Update the state as in Line 7 of Algorithm 1.
9: **Round** 3 **(cleanup).** Move along the lattice path of Lemma 5.1(i) through the occupied product boxes on the grid of product boxes of cubes of side H. On first arrival at an occupied $I \times J$, visit the required configurations in it by Lemma 5.1(i).
10: Output as in Line 10 of Algorithm 1.

Definition 6.2 (Cube and Key Samples)**.** A *cube sample* is a sample whose partition consists of the cubes of side H. A *key sample* is a sample on the partition into single positions.

Take $a_0 > a_1$ from Lemma 2.5 at $r = 3$, and let $b_0 = n^{a_0}$, $A_\star = nb_0$, and $H = \lceil n^{(1-a_0)/D} \rceil$. Let $p = n^{-a_1}$ and $\sigma = n^{-(a_0-a_1)}$. Both rates lie in $(0, 1)$. The rate p is for the tour of the cleanup, since a cube sample leaves dispersion about n/p, and the rate σ is for its visits, since a key sample leaves squared mass about n/σ. The minimum corner of a cube is its point with all coordinates smallest. For the family entering round 2, let P_J be the number of cubes I with $Q_{I,J} > 0$.

The next lemma bounds the route of the cube sample for any family.

Lemma 6.3. *Let a family of disjoint segments have squared mass A, let $1 \leq H \leq n^{1/d}$, and let $p \in (0, 1]$. The route of Lines 3–5 of Algorithm 3 visits the required configurations of a cube sample of side H and rate p, and its expected movement is*

$$O_d\big(p\, n^{2/d} H^{-1} A^{1-1/d} + p\, H\, A + n\big).$$

Proof. We bound in turn the moves of the second head between cubes, the paths of the first head, and the visits inside product boxes. There are at most $(2n^{1/d}/H)^d = 2^d n/H^d$ cubes. The second head moves through them in serpentine order at $O_d(H)$ per cube, which totals $O_d(n/H^{d-1}) = O_d(n)$.

Fix a selected cube J with $P_J \geq 1$. By Lemma 5.1(i) on the grid of cubes, the path of the first head has $O_d\big((n^{1/d}/H) P_J^{1-1/d}\big)$ steps, each of movement $O_d(H)$. Here $P_J \geq 1$ absorbs the additive term, and also the $O_d(n^{1/d})$ movement of the first head from the previous path to this one. So the first head moves $O_d\big(n^{1/d} P_J^{1-1/d}\big)$ for J. Each J is selected with probability p. Moreover, $\sum_J P_J = P \leq \sum_G |\mathcal{P}_H(G)|^2 \leq A$ by Equation (5.1) and $|\mathcal{P}_H(G)| \leq |G|$. By the concavity of $t \mapsto t^{1-1/d}$ over the at most $2^d n/H^d$ cubes,

$$\mathbb{E} \sum_{J \text{ selected}} P_J^{1-1/d} \;\leq\; p\Big(\frac{2^d n}{H^d}\Big)^{1/d} A^{1-1/d} \;=\; 2p\, n^{1/d} H^{-1} A^{1-1/d},$$

so the first head moves $O_d\big(p\,n^{2/d}H^{-1}A^{1-1/d}\big)$ in expectation.

Finally, for an occupied $I \times J$ with J selected, at most $Q_{I,J}$ required configurations lie in $I \times J$, so by Lemma 5.1(i) visiting them and returning takes movement $O_d\big(H\,Q_{I,J}^{1-1/D} + H\big) = O_d(H\,Q_{I,J})$, because $Q_{I,J}$ is a positive integer. Since $\sum_{I,J} Q_{I,J} = \sum_G |G|^2 = A$ and each J is selected with probability p, these visits have expected total $O_d(p\,H\,A)$. □

For a family of squared mass A, the key sample has at most σA required configurations in expectation, because each ordered pair of distinct keys in a common segment is required with probability σ. By Lemma 5.1(ii) and Jensen's inequality, Line 7 of Algorithm 3 has expected movement $O_d\big(n^{1/d}(\sigma A)^{1-1/D} + n^{1/d}\big)$.

Theorem 6.4. *For every $d \geq 2$, Algorithm 3 sorts with expected movement $O_d\big(n^{\beta_3^{(d)}}\big)$ on every input, and its output is correct for every outcome of its coins, where $\beta_3^{(d)} = \frac{16d^2-6d+1}{12d^2-6d+1}$. Hence $M_3^{(d)}(n) = O_d\big(n^{\beta_3^{(d)}}\big)$.*

Proof. Write $\beta = \beta_3^{(d)}$. Correctness was shown after Algorithm 1. We bound the rounds in turn. Round 1 has movement $O(n^\beta)$. By Lemma 5.7, its output has squared mass A with $\mathbb{E}[A] \leq 3A_\star$.

For the cube sample in round 2, we take expectations in Lemma 6.3, using Jensen's inequality for $t \mapsto t^{1-1/d}$. The value of H makes the two main terms equal: since $H = \Theta\big(n^{(1-a_0)/D}\big)$, both are

$$O_d\big(p\,n^{2/d}H^{-1}A_\star^{1-1/d} + p\,H\,A_\star\big) = O_d\big(n^{-a_1+1+a_0+(1-a_0)/D}\big) = O_d(n^\beta),$$

where the last step is Equation (2.2) for $j = 1$. For the key sample, $\sigma A_\star = n^{1+a_1}$, so its expected movement is $O_d\big(n^{1/d+(1+a_1)(1-1/D)} + n^{1/d}\big)$. The exponent equals $1 + a_1 + (1 - a_1)/D$, which is β by Equation (2.3).

For round 3, the output of round 2 is obtained from the output of the cube sample by splitting segments, so Lemma 5.8 gives $\mathbb{E}[S_H] \leq 3n/p$. By Equation (5.1), $P \leq S_H$. By Lemma 5.1(i) on the grid of product boxes, the lattice path has $O_d\big((n^{1/d}/H)(P^{1-1/D} + 1)\big)$ steps, each of movement $O_d(H)$. By Jensen's inequality, the path has expected movement $O_d\big(n^{1/d}(n/p)^{1-1/D} + n^{1/d}\big)$. Since $n/p = n^{1+a_1}$, this is $O_d(n^\beta)$ by the computation for the key sample. By Lemma 5.1(i) and $Q_{I,J} \geq 1$, the visits inside the occupied product boxes take movement $O_d\big(H \sum_{I,J} Q_{I,J}\big)$, and $\sum_{I,J} Q_{I,J}$ is the squared mass of the output of round 2. This output is obtained from the output of the key sample by splitting segments, so Lemma 5.8 for the partition into single positions shows that its squared mass has expectation at most $3n/\sigma$. Hence the visits have expected movement $O_d(Hn/\sigma) = O_d\big(n^{1+(1-a_0)/D+a_0-a_1}\big) = O_d(n^\beta)$, where the last step is Equation (2.2) for $j = 1$.

Summing over the three rounds gives $\mathbb{E}[\text{Move}] = O_d(n^\beta)$. □

6.3 Multiscale Refinement for $r \geq 4$

At $r = 3$ the refinement round is followed by the cleanup, whose cost is controlled by one dispersion and the squared mass. For $r \geq 4$ a refinement round can be followed by another one, whose route moves at every box size (Lemma 6.6), so its cost depends on the dispersion of its input at all these sizes. The analysis therefore keeps an upper bound on the expected dispersion for each box size, and the rates of each round are chosen in advance from this bound.

In this subsection $d \geq 2$ and $r \geq 4$ are fixed and $\beta = \beta_r^{(d)}$. A scale here means a spatial scale: x and y denote volume exponents of boxes, positions are written u and v, and the rank scales enter only through the exponents a_j. Let $c = 1/d$, and recall that $\delta = \beta - 1$ and $a_0 = 1 - \delta$.

Definition 6.5 (Scales and Samples at a Scale)**.** (1) Halving every axis i times, for $0 \leq i \leq \lfloor \log_2 m \rfloor$, partitions $[m]^d$ into 2^{id} boxes with sides $\lfloor m/2^i \rfloor$ and $\lceil m/2^i \rceil$. Such a box is a *scale-x box*, where $n^x = (m/2^i)^d$, and x is its *spatial scale*. We call these x *dyadic*, and also $x = 0$, whose scale-0 boxes are the single positions. These partitions are nested, and a box contains at most 2^d boxes of the next finer partition. There are $B_x = n^{1-x}$ scale-x boxes, with sides at most $2n^{x/d}$, and there are $O(\log n)$ dyadic x. We write $\mathcal{P}_x(G)$ and S_x for the partition into scale-x boxes. Thus S_0 is the squared mass, and $S_x \leq \min\{S_0, nB_x\}$ because $|\mathcal{P}_x(G)| \leq \min\{|G|, B_x\}$.

(2) A *sample at scale y* is a sample on the partition into scale-y boxes. Its required configurations are the set W_y of pairs (u, v) with $u \neq v$ and v a pivot of this sample in the segment of u. For a set W of configurations, $N_x(W)$ is the number of product boxes of two scale-x boxes that contain a configuration of W. Thus $N_x(W) \leq \min\{|W|, B_x^2\}$.

Lemma 6.6. *A set W of configurations is visited by a route that starts at the current configuration and has movement $O_d\big(n^{1/d} + \sum_x n^{cx} N_x(W)\big)$, where the sum ranges over the dyadic x. The route can be computed from W in polynomial time.*

Proof. The product boxes that contain a configuration of W form a tree under inclusion, with the root $[m]^D$. A box has at most 2^D children, and the leaves are the single configurations of W. Mark one configuration of W in each node, and traverse the tree in depth-first order, moving between marked configurations. Each edge is crossed twice. A crossing into or out of a child of a scale-x node moves $O_d(n^{cx})$, because both marked configurations lie in this node, and the $N_x(W)$ nodes of scale x have at most $2^D N_x(W)$ children. So the traversal visits every configuration of W and moves $O_d\big(\sum_x n^{cx} N_x(W)\big)$, and it moves $O_d(n^{1/d})$ from the current configuration to the mark of the root. □

For the required configurations of a sample, the next lemma bounds N_x through the dispersion S_x.

Lemma 6.7. *Fix a family of disjoint segments and draw a sample at scale y with rate p. For every dyadic x, $\mathbb{E}\big[N_x(W_y)\big] \leq p\, S_x$.*

Proof. A configuration of W_y consists of a key of a segment G and a pivot of G. So a segment G contributes to $N_x(W_y)$ only through product boxes $I \times J$ such that I meets G and J contains a pivot of G. Let $t_x(G)$ be the number of scale-x boxes that contain a pivot of G. There are at most $|\mathcal{P}_x(G)|\, t_x(G)$ such product boxes. If $x \leq y$, a scale-x box that meets G contains a pivot of G exactly when the scale-y box containing it is selected, so $\mathbb{E}[t_x(G)] = p\,|\mathcal{P}_x(G)| \leq p\,|G|$. If $x \geq y$, a scale-x box that contains a pivot of G contains a selected scale-y box that meets G, so $\mathbb{E}[t_x(G)] \leq p\,|\mathcal{P}_y(G)| \leq p\,|G|$. In both cases $\mathbb{E}\big[|\mathcal{P}_x(G)|\, t_x(G)\big] \leq p\,|G|\,|\mathcal{P}_x(G)|$, and summing over the segments gives the bound. □

The rate alone gives one more bound, for every family. A product box $I \times J$ contains a configuration of W_y only if J contains a pivot. If $x \leq y$, this requires the scale-y box containing J to be selected, so $\mathbb{E}[N_x(W_y)] \leq p\, B_x^2$. If $x \geq y$, it requires J to contain a selected scale-y box, and $p\, B_y$ scale-y boxes are selected in expectation, so $\mathbb{E}[N_x(W_y)] \leq p\, B_x B_y$. In both cases,

$$\mathbb{E}[N_x(W_y)] \;\leq\; p\, n^{2-x-\min\{x,y\}}. \tag{6.1}$$

Algorithm 4 The multiscale algorithm

Parameters: $d \geq 2$; $r \geq 4$; the functions $f_0, \dots, f_{r-2}$ as above.

1: **Round 1 (splitter).** Lines 1–3 of Algorithm 1.
2: **for** $j = 1, \dots, r-2$ **do**
3: **Round** $j+1$ **(refinement).** For every dyadic $y \leq x_\star$, draw a sample at scale y with rate $n^{-f_j(y)}$, independently.
4: For every such y, visit W_y by the route of Lemma 6.6, and join these routes end to end.
5: Update the state as in Line 7 of Algorithm 1.
6: **end for**
7: **Round** r **(cleanup).** Visit the required configurations by the route of Lemma 6.6.
8: Output as in Line 10 of Algorithm 1.

Only the scales $x \leq x_\star$, where $x_\star = a_0/(2-c)$, require control of the segments. For dyadic $x \geq x_\star$, $n^{cx} B_x^2 = n^{2-(2-c)x} \leq n^{2-a_0} = n^\beta$, so each of these scales contributes at most n^β to the bound of Lemma 6.6 for any set of configurations.

Definition 6.8 (Profiles)**.** A *profile* of a random family of segments is a function $f\colon [0, x_\star] \to \mathbb{R}$ with $\mathbb{E}[S_x] \leq 3n^{1+f(x)}$ at every dyadic $x \leq x_\star$. For a function f on $[0, x_\star]$, let $\varphi_f(x) = f(x) + cx - \delta$ and $\psi_y(x) = a_0 - (1-c)x - \min\{x, y\}$, and let

$$(\Phi f)(y) = \sup_{0 \leq x \leq x_\star} \min\bigl\{\varphi_f(x),\, \psi_y(x)\bigr\}.$$

The functions φ_f and ψ_y give two ways to keep the contribution of a scale x to the movement of a sample at scale y at $O(n^\beta)$: through the dispersion, with rate at most $n^{-\varphi_f(x)}$, or through the rate alone, with rate at most $n^{-\psi_y(x)}$. So $n^{-(\Phi f)(y)}$ is the largest rate for which one of the two applies at every $x \leq x_\star$ (Lemma 6.9).

We use two facts about $x_\star$, in Lemma 6.10 and Theorem 6.11. First,

$$a_{r-2} = \delta - c\, x_\star. \tag{6.2}$$

Indeed, Lemma 2.5 gives $\delta(1-q^r) = 1-q$ and $a_{r-2} = q^{r-1}\delta$, and $c = 2(1-q)$ and $2-c = 2q$, so

$$\delta - c\, x_\star = \delta - \frac{c\,(1-\delta)}{2-c} = \frac{\delta - (1-q)}{q} = \frac{\delta\, q^r}{q} = a_{r-2}.$$

Second, $\delta \leq x_\star$. This inequality is equivalent to $(1-q^r)/(1-q) \geq 1 + 2q$, which holds for $r \geq 4$ because $q \geq \frac{3}{4}$ gives $q^2 + q^3 \geq q$.

Let $f_0(x) = \min\{a_0, 1-x\}$ and $f_j = \Phi f_{j-1}$ for $1 \leq j \leq r-2$. Thus f_0 is the bound that the splitters and the trivial bound $S_x \leq n^{2-x}$ give, and f_j is the bound that we keep after round $j+1$. Algorithm 4 is the instance of Algorithm 1 that draws one sample per dyadic scale.

The next lemma is the inductive step of the analysis: it bounds the movement of one refinement round and the profile of its output.

Lemma 6.9. *Let f be a profile of the family entering a refinement round, and let $(\Phi f)(y) \geq 0$ at every dyadic $y \leq x_\star$. Suppose that the round draws, for every such y, a sample at scale y with rate $n^{-(\Phi f)(y)}$, independently, and visits W_y by the route of Lemma 6.6. Then the round has expected movement $O_d(n^\beta \log^2 n)$, and Φf is a profile of its output.*

Proof. We bound the contribution of each scale to each sample, and then the profile of the output. Fix a dyadic $y \le x_\star$, let $p = n^{-(\Phi f)(y)}$, and fix a dyadic x. We show that $n^{cx}\,\mathbb{E}[N_x(W_y)] \le 3n^\beta$. If $x \ge x_\star$, this follows from $N_x(W_y) \le B_x^2$ and $n^{cx}B_x^2 \le n^\beta$. Let $x \le x_\star$. By the definition of Φ, $p \le n^{-\min\{\varphi_f(x),\psi_y(x)\}}$, so $p \le n^{-\varphi_f(x)}$ or $p \le n^{-\psi_y(x)}$. In the first case, Lemma 6.7 and the profile give

$$n^{cx}\,\mathbb{E}[N_x(W_y)] \;\le\; n^{cx}\,p\,\mathbb{E}[S_x] \;\le\; 3\,n^{cx-f(x)-cx+\delta+1+f(x)} \;=\; 3n^\beta.$$

In the second case, Equation (6.1) gives

$$n^{cx}\,\mathbb{E}[N_x(W_y)] \;\le\; n^{cx-a_0+(1-c)x+\min\{x,y\}+2-x-\min\{x,y\}} \;=\; n^{2-a_0} \;=\; n^\beta.$$

By Lemma 6.6, the route of this sample has expected movement $O_d(n^{1/d} + n^\beta \log n)$, since there are $O(\log n)$ dyadic x. There are $O(\log n)$ samples, which gives the bound on the movement.

For the output, Lemma 5.8 shows that the refined segments of the sample at scale y satisfy $\mathbb{E}[S_y] \le 3n/p = 3n^{1+(\Phi f)(y)}$, whatever the entering family. The output of the round is obtained from them by splitting segments, so it satisfies the same bound. □

The cleanup needs $\varphi_{f_{r-2}} \le 0$ on $[0, x_\star]$, which follows from two facts given by the next lemma: the value of f_{r-2} at $x_\star$ and the monotonicity of $\varphi_{f_{r-2}}$.

Lemma 6.10. *For $1 \le j \le r-2$, the function f_j is nonincreasing, φ_{f_j} is nondecreasing, and $f_j = a_j$ on $[\xi_j, x_\star]$, where $\xi_j = (a_0 - a_j)/(2-c)$. Consequently $f_j \ge a_j$.*

Proof. We use induction on j. For $j = 1$, recall that $\delta \le x_\star$. On $[0, \delta]$ we have $f_0 = a_0$, so $\varphi_{f_0}(x) = 1 - 2\delta + cx$ increases. On $[\delta, x_\star]$ we have $f_0(x) = 1 - x$, so $\varphi_{f_0}(x) = a_0 - (1-c)x \ge \psi_y(x)$, and the minimum in Φ is ψ_y, which decreases. So the supremum is attained on $[0, \delta]$ where φ_{f_0} and ψ_y cross. For $y \ge \delta/2$ they cross at $x = \delta/2$, and for $y \le \delta/2$ they cross at $x = \delta - y$. This gives

$$f_1(y) = \begin{cases} 1 - (2-c)\delta - cy & \text{for } y \le \delta/2, \\ 1 - 2\delta + c\delta/2 & \text{for } y \ge \delta/2. \end{cases}$$

Since $q = 1 - c/2$, the second value equals $q\,a_0 + c/2 - \delta = a_1$, and hence $\xi_1 = \delta/2$. The formula shows that f_1 is nonincreasing and φ_{f_1} is nondecreasing.

For the step, let $f = f_j$, $a = a_j$, and $\xi = \xi_j$, with $j \le r-3$. Since f is nonincreasing and φ_f is nondecreasing, φ_f is c-Lipschitz. The function Φf is nonincreasing because ψ_y decreases in y. To see that $\varphi_{\Phi f}$ is nondecreasing, let $z > y$, let $\Delta = z - y$, and for $x \in [0, x_\star]$ let $x' = \max\{x - \Delta, 0\}$. Then $\varphi_f(x') \ge \varphi_f(x) - c\Delta$. Moreover $\psi_z(x') \ge \psi_y(x) - c\Delta$: for $x < \Delta$ we have $\psi_z(0) = a_0 \ge \psi_y(x)$, and for $x \ge \Delta$ this follows from $\min\{x - \Delta, z\} \le \min\{x, y\} + \Delta$. Hence $(\Phi f)(z) \ge (\Phi f)(y) - c\Delta$, which is the claim.

For the value on the interval, let $a' = q\,a + c/2 - \delta$ and $\xi' = (a_0 - a')/(2-c)$. Since $c/2 = 1/D$, Equation (2.2) gives $a' = a_{j+1}$, so $a > a' > 0$ and $\xi < \xi' < x_\star$. Thus $f(\xi') = a$, and a direct computation from $a_0 = 1 - \delta$ and $q + c/2 = 1$ gives $a + c\xi' - \delta = a'$. Hence, for $y \ge \xi'$,

$$\varphi_f(\xi') = a' = a_0 - (2-c)\xi' = \psi_y(\xi').$$

Since $\varphi_f \le a'$ to the left of ξ' and $\psi_y \le a'$ to the right of it, we get $(\Phi f)(y) = a'$ for $y \in [\xi', x_\star]$.

Finally, $f_j \ge f_j(x_\star) = a_j$ because f_j is nonincreasing. □

Since $f_j \geq a_j > 0$, the rates of Algorithm 4 lie in $(0,1)$, and Lemma 6.9 applies to round $j+1$ with $f = f_{j-1}$.

Theorem 6.11. *For every fixed $d \geq 2$ and $r \geq 4$, Algorithm 4 sorts with expected movement $O_{d,r}\big(n^{\beta_r^{(d)}} \log^2 n\big)$ on every input, and its output is correct for every outcome of its coins. Hence $M_r^{(d)}(n) = O_{d,r}\big(n^{\beta_r^{(d)}} \log^2 n\big)$.*

Proof. Correctness was shown after Algorithm 1. We bound the rounds in turn. Round 1 has movement $O(n^\beta)$. By Lemma 5.7 with $s = \lceil n/b_0 \rceil$, its output satisfies $\mathbb{E}[S_0] \leq 3nb_0 = 3n^{1+a_0}$. Since $S_x \leq S_0$ and $S_x \leq n^{2-x}$, the function f_0 is a profile of this family.

By Lemmas 6.9 and 6.10, for $1 \leq j \leq r-2$, round $j+1$ has expected movement $O_d(n^\beta \log^2 n)$, and f_j is a profile of its output.

It remains to bound the cleanup, whose required configurations we call W. At a dyadic x, a product box of two scale-x boxes contains a configuration of W only if a segment meets both boxes, so $N_x(W) \leq \sum_G |\mathcal{P}_x(G)|^2 \leq S_x$. Also $N_x(W) \leq B_x^2$. Let $x \leq x_\star$. Since $\varphi_{f_{r-2}}$ is nondecreasing and $f_{r-2}(x_\star) = a_{r-2}$, Equation (6.2) gives

$$\varphi_{f_{r-2}}(x) \;\leq\; \varphi_{f_{r-2}}(x_\star) \;=\; a_{r-2} + c\,x_\star - \delta \;=\; 0.$$

That is, $cx + f_{r-2}(x) \leq \delta$, so $n^{cx}\,\mathbb{E}[S_x] \leq 3n^{cx+1+f_{r-2}(x)} \leq 3n^\beta$. For $x \geq x_\star$, we have $n^{cx} B_x^2 \leq n^\beta$. By Lemma 6.6, the cleanup has expected movement $O_d(n^\beta \log n)$, since there are $O(\log n)$ dyadic x.

Summing over the r rounds gives $\mathbb{E}[\mathrm{Move}] = O_{d,r}(n^\beta \log^2 n)$. □

Proof of Theorem 1.1. For finite r, the lower bound is Theorem 4.1. For the upper bounds, Theorem 5.11 treats $d = 1$. For $d \geq 2$, Proposition 6.1 treats $r \leq 2$, Theorem 6.4 treats $r = 3$, and Theorem 6.11 treats $r \geq 4$. These algorithms are explicit and run in polynomial time. Those for $r = 1$ are deterministic routes, and the others are instances of Algorithm 1. This gives $M_r^{(d)}(n) = \widetilde{\Theta}\big(n^{\beta_r^{(d)}}\big)$ for finite r. For $r = \infty$, Propositions 4.5 and 5.3 give $M_\infty^{(d)}(n) = \Theta_d\big(n^{1+1/(2d)}\big)$, and the algorithm of Proposition 5.3 is deterministic, explicit, and runs in polynomial time. The order is exact on the line by Theorems 4.12 and 5.11 and for $r \leq 2$ by Proposition 4.4 together with Theorem 5.11 and Proposition 6.1. □

7 Multiple Heads

So far the model has two heads. With $k \geq 2$ heads on $[m]^d$, a configuration is a point of $([m]^d)^k$, a route is a sequence of configurations at consecutive ℓ_1 distance 1, and the outcome at a configuration is the rank order of the distinct keys under the heads. Rounds, correctness, and the complexity $M_r^{(k,d)}(n)$ are defined as in Definition 2.1, so $M_r^{(2,d)} = M_r^{(d)}$. Extra heads can only help: an algorithm may leave them at the start and ignore what they read, so $M_r^{(k,d)} \leq M_r^{(d)}$.

A fixed number of heads nevertheless changes the complexity by at most a constant factor. Two keys are compared at every configuration that places two heads on them. So all that a k-head algorithm learns is read along the $\binom{k}{2}$ projections of its route to pairs of heads, and each projection, with consecutive repetitions deleted, is a route in the same comparison grid $[m]^D$.

Proposition 7.1. *For every fixed $k \geq 2$ and d, and every $r \in \{1,2,\dots\} \cup \{\infty\}$, $M_r^{(k,d)}(n) = \Theta_{d,r,k}\big(M_r^{(d)}(n)\big)$.*

Proof. The upper bound on $M_r^{(k,d)}$ was shown above. For the lower bound, a two-head algorithm simulates a given k-head algorithm one pair of heads at a time when r is finite, and for $r = \infty$ we bound the movement directly. Let r be finite. A round of the k-head algorithm commits a route $z_0, \dots, z_L$. Let $L^{(i)}$ be the movement of head i on this route, so that $L = \sum_i L^{(i)}$. For a pair $i < j$, let R_{ij} be the sequence of positions of heads i and j along the route, with consecutive repetitions deleted. It is a route in $[m]^D$ of movement at most $L^{(i)} + L^{(j)}$. In one round of its own, the simulator runs the routes R_{ij} in turn. These runs have total movement at most $\sum_{i<j}(L^{(i)} + L^{(j)}) = (k-1)L$, because each head lies in $k-1$ pairs. The moves from one run to the next add $O_{d,k}(n^{1/d})$ per round. The pairwise outcomes determine the rank order of the keys under the heads at every configuration of the route, so the simulator knows all outcomes of the round and can compute the next route. Hence

$$M_r^{(d)}(n) \;\le\; (k-1)\, M_r^{(k,d)}(n) + O_{d,r,k}\big(n^{1/d}\big).$$

Since $M_r^{(d)}(n) = \Omega_d\big(n^{1+1/(2d)}\big)$ by Proposition 4.5 and $1 + 1/(2d) > 1/d$, the additive term is $o\big(M_r^{(d)}(n)\big)$, and the proposition follows.

The simulation adds a term in every round, so for $r = \infty$ we argue directly and bound the movement of an algorithm with any finite number of rounds. By Yao's principle, fix a deterministic k-head algorithm that is correct on every input, and let the input be uniform. Let $\rho = \Theta_d(n^{1/(2d)})$ be the radius of Lemma 3.3 at scale 2, whose cells join the ranks $2i-1$ and $2i$. With probability at least $7/8$, there is a set $W \subseteq \mathcal{C}_2$ of at least $\frac{1}{2}\lfloor n/2 \rfloor$ cells that are pairwise more than ρ apart. For every cell $\{u, v\}$ of W, some configuration of the execution places two heads on u and v. Otherwise, exchanging their two consecutive ranks changes no outcome, so the algorithm gives the same output on two inputs with different orders. For a pair $i < j$, let $W_{ij} \subseteq W$ be the set of cells that heads i and j visit together, and let $\widehat{L}^{(i)}$ be the total movement of head i. By Observation 2.3(b), the positions of heads i and j over the whole execution, with consecutive repetitions and diagonal configurations deleted, form a sequence of cells of dist-length at most $\widehat{L}^{(i)} + \widehat{L}^{(j)}$. By Fact 3.2, if $|W_{ij}| \ge 2$, then $\widehat{L}^{(i)} + \widehat{L}^{(j)} \ge \frac{1}{2}|W_{ij}|\rho$. Summing over the pairs, and noting that the pairs with $|W_{ij}| \le 1$ cover at most $\binom{k}{2}$ cells, gives

$$(k-1)\,\mathrm{Move} \;=\; \sum_{i<j}\big(\widehat{L}^{(i)} + \widehat{L}^{(j)}\big) \;\ge\; \frac{\rho}{2}\Big(\frac{1}{2}\Big\lfloor\frac{n}{2}\Big\rfloor - \binom{k}{2}\Big)$$

on this event. Hence $\mathbb{E}[\mathrm{Move}] = \Omega_{d,k}\big(n^{1+1/(2d)}\big)$, with a constant that does not depend on the number of rounds, so $M_\infty^{(k,d)}(n) = \Omega_{d,k}\big(n^{1+1/(2d)}\big)$. With Proposition 5.3, this is $\Omega_{d,k}\big(M_\infty^{(d)}(n)\big)$. □

8 Discussion

In the interior $d \ge 2$, $3 \le r < \infty$, the two bounds of Theorem 1.1 differ by a factor $O_{d,r}(\log^3 n)$, and this gap is the one question the theorem leaves open. The lower bound loses a factor $\log n$ throughout the interior, and the upper bound loses a factor $\log^2 n$ for $r \ge 4$ and nothing at $r = 3$. We conjecture that both losses can be removed.

Conjecture 8.1. *For every $d \ge 2$ and $3 \le r < \infty$, $M_r^{(d)}(n) = \Theta_{d,r}\big(n^{\beta_r^{(d)}}\big)$.*

AI Methodology Statement

The author used ChatGPT and Codex from OpenAI and Claude Code from Anthropic to help organize and polish the exposition of this paper, based on the author's own drafts. The same tools assisted in surveying the related literature, in understanding the current state of research in this area, in flagging possible errors in draft proofs, and in suggesting simpler proofs of some statements. The author checked, revised, and finalized every statement, proof, and reference. The author takes full responsibility for the content of the submission.

References

[AA88] Noga Alon and Yossi Azar. Sorting, approximate sorting, and searching in rounds. *SIAM Journal on Discrete Mathematics*, 1(3):269–280, 1988.

[AAV86] Noga Alon, Yossi Azar, and Uzi Vishkin. Tight complexity bounds for parallel comparison sorting. In *27th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 502–510, 1986.

[AV88] Alok Aggarwal and Jeffrey Scott Vitter. The input/output complexity of sorting and related problems. *Communications of the ACM*, 31(9):1116–1127, 1988.

[BABP03] Amir M. Ben-Amram, Omer Berkman, and Holger Petersen. Element distinctness on one-tape Turing machines: a complete solution. *Acta Informatica*, 40(2):81–94, 2003.

[BB90] Béla Bollobás and Graham Brightwell. Parallel selection with high probability. *SIAM Journal on Discrete Mathematics*, 3(1):21–31, 1990.

[BC82] Allan Borodin and Stephen A. Cook. A time-space tradeoff for sorting on a general sequential model of computation. *SIAM Journal on Computing*, 11(2):287–297, 1982.

[Bea91] Paul Beame. A general sequential time-space tradeoff for finding unique elements. *SIAM Journal on Computing*, 20(2):270–277, 1991.

[BG89] Dimitris Bertsimas and Michelangelo Grigni. Worst-case examples for the spacefilling curve heuristic for the Euclidean traveling salesman problem. *Operations Research Letters*, 8(5):241–244, 1989.

[BHH59] Jillian Beardwood, J. H. Halton, and J. M. Hammersley. The shortest path through many points. *Proceedings of the Cambridge Philosophical Society*, 55(4):299–327, 1959.

[BMP19] Mark Braverman, Jieming Mao, and Yuval Peres. Sorted top-k in rounds. In *32nd Annual Conference on Learning Theory (COLT)*, volume 99 of *Proceedings of Machine Learning Research*, pages 342–382, 2019.

[BR16] Indranil Banerjee and Dana Richards. Sorting under forbidden comparisons. In *15th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, volume 53 of *LIPIcs*, pages 22:1–22:13, 2016. arXiv:1508.03698.

[BT83] Béla Bollobás and Andrew Thomason. Parallel sorting. *Discrete Applied Mathematics*, 6(1):1–11, 1983.

[CFG+02] Moses Charikar, Ronald Fagin, Venkatesan Guruswami, Jon Kleinberg, Prabhakar Raghavan, and Amit Sahai. Query strategies for priced information. *Journal of Computer and System Sciences*, 64(4):785–819, 2002.

[DMS91] Martin Dietzfelbinger, Wolfgang Maass, and Georg Schnitger. The complexity of matrix transposition on one-tape off-line Turing machines. *Theoretical Computer Science*, 82(1):113–129, 1991.

[Few55] L. Few. The shortest path and the shortest road through n points. *Mathematika*, 2(2):141–144, 1955.

[FHLS19] Alireza Farhadi, MohammadTaghi Hajiaghayi, Kasper Green Larsen, and Elaine Shi. Lower bounds for external memory integer sorting via network coding. In *51st*

Annual ACM Symposium on Theory of Computing (STOC), pages 997–1008, 2019. arXiv:1811.01313.

[Fil24] Arnold Filtser. Scattering and sparse partitions, and their applications. *ACM Transactions on Algorithms*, 20(4):30:1–30:42, 2024. arXiv:2001.04447.

[FS89] Michael L. Fredman and Michael E. Saks. The cell probe complexity of dynamic data structures. In *21st Annual ACM Symposium on Theory of Computing (STOC)*, pages 345–354, 1989.

[GHR06] Anupam Gupta, Mohammad T. Hajiaghayi, and Harald Räcke. Oblivious network design. In *17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 970–979, 2006.

[GJ24] Mayank Goswami and Riko Jacob. An algorithm for bichromatic sorting with polylog competitive ratio. In *15th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 287 of *LIPIcs*, pages 56:1–56:17, 2024.

[GK01] Anupam Gupta and Amit Kumar. Sorting and selection with structured costs. In *42nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 416–425, 2001.

[GK05] Anupam Gupta and Amit Kumar. Where's the winner? Max-finding and sorting with metric costs. In *Approximation, Randomization and Combinatorial Optimization (APPROX/RANDOM)*, volume 3624 of *Lecture Notes in Computer Science*, pages 74–85. Springer, 2005.

[HH82] Roland Häggkvist and Pavol Hell. Sorting and merging in rounds. *SIAM Journal on Algebraic and Discrete Methods*, 3(4):465–473, 1982.

[HKK11] Zhiyi Huang, Sampath Kannan, and Sanjeev Khanna. Algorithms for the generalized sorting problem. In *52nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 738–747, 2011.

[HKL06] Mohammad T. Hajiaghayi, Robert Kleinberg, and Tom Leighton. Improved lower and upper bounds for universal TSP in planar metrics. In *17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 649–658, 2006.

[Jai88] Patrick Jaillet. A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Operations Research*, 36(6):929–936, 1988.

[JLN+05] Lujun Jia, Guolong Lin, Guevara Noubir, Rajmohan Rajaraman, and Ravi Sundaram. Universal approximations for TSP, Steiner tree, and set cover. In *37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 386–395, 2005.

[KN21] William Kuszmaul and Shyam Narayanan. Stochastic and worst-case generalized sorting revisited. In *62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1056–1067, 2021. arXiv:2111.07222.

[Kra26] Cosmas Kravaris. Lower bounds for the universal TSP on the plane. In *2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1–9, 2026. arXiv:2412.16448.

[LL04] Zongpeng Li and Baochun Li. Network coding: the case of multiple unicast sessions. In *42nd Allerton Conference on Communication, Control, and Computing*, 2004.

[LO94] Alejandro López-Ortiz. New lower bounds for element distinctness on a one-tape Turing machine. *Information Processing Letters*, 51(6):311–314, 1994.

[MNSW98] Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. *Journal of Computer and System Sciences*, 57(1):37–49, 1998.

[MS86] Wolfgang Maass and Georg Schnitger. An optimal lower bound for Turing machines with one work tape and a two-way input tape. In *Structure in Complexity Theory*, volume 223 of *Lecture Notes in Computer Science*, pages 249–264. Springer, 1986.

[PB89] Loren K. Platzman and John J. Bartholdi, III. Spacefilling curves and the planar travelling salesman problem. *Journal of the ACM*, 36(4):719–737, 1989.

[Pet08] Holger Petersen. Element distinctness and sorting on one-tape off-line Turing machines. In *34th Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, volume 4910 of *Lecture Notes in Computer Science*, pages 406–417. Springer, 2008.

[Pip87] Nicholas Pippenger. Sorting and selecting in rounds. *SIAM Journal on Computing*, 16(6):1032–1038, 1987.

[Ste81] J. Michael Steele. Subadditive Euclidean functionals and nonlinear growth in geometric probability. *The Annals of Probability*, 9(3):365–376, 1981.

[Tho83] Clark D. Thompson. The VLSI complexity of sorting. *IEEE Transactions on Computers*, C-32(12):1171–1184, 1983.

[TK77] C. D. Thompson and H. T. Kung. Sorting on a mesh-connected parallel computer. *Communications of the ACM*, 20(4):263–271, 1977.

[Val75] Leslie G. Valiant. Parallelism in comparison problems. *SIAM Journal on Computing*, 4(3):348–355, 1975.

[Yao77] Andrew Chi-Chih Yao. Probabilistic computations: toward a unified measure of complexity. In *18th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 222–227, 1977.

[YDI+15] Katsuhisa Yamanaka, Erik D. Demaine, Takehiro Ito, Jun Kawahara, Masashi Kiyomi, Yoshio Okamoto, Toshiki Saitoh, Akira Suzuki, Kei Uchizawa, and Takeaki Uno. Swapping labeled tokens on graphs. *Theoretical Computer Science*, 586:81–94, 2015.

[Yuk98] Joseph E. Yukich. *Probability Theory of Classical Euclidean Optimization Problems*, volume 1675 of *Lecture Notes in Mathematics*. Springer, 1998.