SIAM J. DISC. MATH.
Vol. 1, No. 3, August 1988

001

SORTING, APPROXIMATE SORTING, AND SEARCHING IN ROUNDS*

NOGA ALON† AND YOSSI AZAR‡

Abstract. The worst case number of comparisons needed for sorting or selecting in rounds is considered. The following results are obtained.

(a) For every fixed $k \geqq 2$, $\Omega(n^{1+1/k}(\log n)^{1/k})$ comparisons are required to sort n elements in k rounds. ($O(n^{1+1/k}\log n)$ are known to be sufficient.) This improves the previous known bounds by a factor of $(\log n)^{1-1/k}$, which separates deterministic algorithms from randomized ones, as there are randomized algorithms whose expected number of comparisons is $O(n^{1+1/k})$.

(b) For every fixed $k \geqq 2$, $\Omega(n^{1+1/(2^k-1)}(\log n)^{2/(2^k-1)})$ comparisons are required to select the median from n elements in k rounds. ($O(n^{1+1/(2^k-1)}(\log n)^{2-2/(2^k-1)})$ are known to be sufficient.) This improves the previous known bounds by a factor of $(\log n)^{2/(2^k-1)}$ and separates the problem of finding the median from that of finding the minimum, as $O(n^{1+1/(2^k-1)})$ comparisons suffice for finding the minimum.

(c) We show that "approximate sorting" in one round requires asymptotically more than $c \cdot n \log n$ comparisons, for every constant c, and can be done in $O(n \log n \log \log n)$ comparisons. This settles a problem raised by Rabin.

Key words. sorting in rounds, searching, approximate sorting, parallel comparison algorithms

AMS(MOS) subject classification. 68E05

1. Introduction. Sorting and selecting are two of the principal problems in computer science. As mentioned in [Kn73], existing computers devote approximately one quarter of their time to sorting. The advent of parallel computers stimulated intensive research of sorting and selecting with respect to various models of parallel computation. Extensive lists of references that recorded this activity are given in [Ak85] and [BHe85].

Most of the fastest serial and parallel sorting or selecting algorithms are based on binary comparisons. In these algorithms the number of comparisons is typically the primary measure of time complexity. Any lower bound on the number of comparisons required for a problem, clearly implies a time lower bound for such algorithms. In the present paper, we restrict our attention to a parallel comparison model, introduced by Valiant [Va75] (see also [BHo82]), where only comparisons are counted. In measuring time complexity within this model, we do not count steps in which communication among the processors, movement of data, and memory addressing are performed. We also avoid counting steps in which consequences are deduced from comparisons that were performed. Note that our lower bounds apply to all comparison-based algorithms in any model.

In a serial decision tree model, we wish to minimize the number of comparisons. The goal of an algorithm in a parallel comparison model is to minimize the number of comparison rounds as well as the total number of comparisons performed.

We consider three problems: sorting, "approximate sorting," and selecting.

1.1. Sorting in rounds. Let k stand for the number of comparison rounds (time) of an algorithm in the parallel comparison model. Let $c(k, n)$ denote the *minimum total* number of comparisons required to sort any n elements in k rounds (over all possible algorithms).

---

* Received by the editors March 15, 1987; accepted for publication (in revised form) February 5, 1988.

† School of Mathematical Sciences, Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv, Israel.
This research was supported in part by an Allon Fellowship, by a grant from Bat Sheva de Rothschild Foundation, and by the Fund for Basic Research Administered by the Israel Academy of Sciences.

‡ School of Mathematical Sciences, Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv, Israel.

Clearly $c(1, n) = \binom{n}{2}$. This is since any sorting algorithm that works in one round must perform all comparisons. Otherwise, suppose that a dispensed comparison is between two successive elements in the sorted order; the algorithm will clearly fail to distinguish their order. On the other hand, performing all comparisons simultaneously yields a one round algorithm with that many comparisons.

In general, it is known that for every fixed k

$$\Omega(n^{1+1/k}) \leq c(k, n) \leq O(n^{1+1/k} \log n).$$

The upper bound is due to Bollobás and Hell [BHe85] (see also [Pi87]). The lower bound is due to Häggkvist and Hell [HH81] (see also [AV87], [AAV86] for the case of not necessarily constant k).

Even the first nontrivial case, that of sorting n elements in two rounds, received considerable attention. Häggkvist and Hell [HH81] showed that

$$\frac{1}{8} n^{3/2} - \frac{1}{2} n \leq c(2, n) = O(n^{5/3} \log n).$$

Bollobás and Thomason [BT83] improved it and showed that

$$c_1 \cdot n^{3/2} \leq c(2, n) = O(n^{3/2} \log n)$$

for any $c_1 < \sqrt{2/3}$, if $n > n(c_1)$.

Explicit algorithms for sorting in two rounds with $o(n^2)$ comparisons are given in [Pi85], [Al85], and [Pi87].

Here we (slightly) improve both bounds and prove Theorem 1.1.

THEOREM 1.1.

$$\Omega(n^{3/2}\sqrt{\log n}) \leq c(2, n) \leq O\left[n^{3/2} \frac{\log n}{\sqrt{\log \log n}}\right].$$

We also prove the following theorem.

THEOREM 1.2. *For every fixed* $k \geq 2$

$$c(k, n) = \Omega(n^{1+1/k}(\log n)^{1/k}).$$

This improves on Häggkvist and Hell's lower bound mentioned above. Notice that the only difference between our improved lower bound and the previously known one, is an extra factor of $(\log n)^{1/k}$. Nevertheless, this is precisely the factor that separates the asymptotic behavior of the best randomized algorithm (which is $\Theta(n^{1+1/k})$; see [AA87], [AAV86]) from that of the best deterministic one.

1.2. Approximate sorting in one round. An algorithm that approximately sorts n elements in one round with p comparisons is a set of p pairs of element, $(a_i, b_i)_{i=1}^{p}$ from the set V of n elements we have to sort, such that for each possible set of answers for the p questions "is $a_i < b_i$," the relative order of all but $o(n^2)$ of the pairs of elements of V will be known.

Note that such an algorithm is simply the first round of a two rounds sorting algorithm that uses only $o(n^2)$ comparisons in the second round. Let $a(n)$ denote the minimum p such that an approximate sorting algorithm in one round with p comparisons exists. Bollobás and Rosenfeld studied these algorithms in [BR81] (also see [BiHe85]) and their results imply that for every fixed $\epsilon > 0$, $a(n) = o(n^{1+\epsilon})$.

M. Rabin (cf. [BHe85]) asked whether $a(n) = O(n \log n)$. The next proposition shows that this is false.

THEOREM 1.3. (i) $\lim_{n\to\infty} a(n)/n \log n \to \infty$. *More precisely, for every* $\epsilon > 0$, *any two rounds sorting algorithm that uses at most* ϵn^2 *comparisons in the second round must use* $\Omega((1/\epsilon) n \log n)$ *comparisons in the first round.*

(ii) $a(n) \leq O(n \log n \log \log n)$.

More precisely, there is a two rounds sorting algorithm that uses

$$O(n \log n \log \log n)$$

comparisons in the first round and $O(n^2/\log \log \log n) = o(n^2)$ comparison in the second round.

Ajtái et al. [AKSS86b] have recently proved, independently of our work, a slightly weaker result than Theorem 1.3. Their bounds are:

$$\Omega(n \log n) \leq a(n) \leq O(n \log n \log \log n w(n))$$

where $w(n)$ is any function that tends to infinity as n tends to infinity. (Note that this lower bound does not suffice to settle Rabin's problem.) More recently, Bollobás and Brightwell [BB87] improved the upper bound and showed that

$$a(n) = O(n \log n \log \log n \cdot w(n)/\log \log \log n),$$

where $w(n)$ is as above.

1.3. Selecting in a fixed number of rounds. Häggkvist and Hell [HH80] showed that $\Omega(n^{1+1/(2^k-1)})$ comparisons are necessary for selection in k rounds and that $O(n^{1+1/(2^k-1)})$ comparisons are sufficient for finding the minimum (or selecting any element of a fixed rank) among n elements in k rounds. Pippenger [Pi87] proved that $O(n^{1+1/(2^k-1)}(\log n)^{2-2/(2^k-1)})$ suffice for selecting the median (or any element of a given rank) in k rounds. A natural problem that arises here is whether the median can be found as efficiently as the minimum. Solving this problem we prove that finding the median in k rounds requires $\Omega(n^{1+1/(2^k-1)}(\log n)^{2/(2^k-1)})$ comparisons. This separates the asymptotic number of comparisons needed for selecting the median from that needed for selecting the minimum. We also slightly improve the best known upper bounds for selection, which are given in [Pi87]. Let $s(k, n)$ denote the minimum total worst case number of comparisons needed to select the median of n elements in k rounds. Our result is the following theorem.

THEOREM 1.4. *For every fixed* $k \geq 2$

$$\Omega(n^{1+1/(2^k-1)}(\log n)^{2/(2^k-1)}) \leq s(k, n)$$
$$\leq O(n^{1+1/(2^k-1)}(\log n)^{2-2/(2^k-1)}/(\log \log n)^{1-1/(2^k-1)}).$$

All our results are proved by combining probabilistic arguments with various results from extremal graph theory. The paper is organized as follows. In § 2 we first prove a lemma about proper vertex colorings of graphs and then deduce from it the lower bounds of Theorems 1.1, 1.2, 1.3, and 1.4. In § 3 we combine certain probabilistic arguments with some of the ideas of [AKSS86a], [BT83], and [Pi87] to obtain all the upper bounds.

2. The lower bounds.

2.1. The parallel computation model. Let V be a set of n elements taken from a totally ordered domain. The *parallel comparison model of computation* allows algorithms that work as follows. The algorithm consists of timesteps called *rounds*. In each round binary comparisons are performed simultaneously. The input for each comparison are two elements of V. The output of each comparison is one of the following two: < or >. Note that we do not allow equality between two elements of V. This can be done without loss of generality, since we define the order between two equal input elements to be the order of their indices. Each item may take part in several comparisons during the same round.

Our discussion uses the following correspondence between each round and a graph. The elements are the vertices. Each comparison to be performed is an undirected edge that connects its input elements. Each computation results in orienting this edge from the largest element to the smallest. Thus in each round we get an acyclic orientation of the corresponding graph, and the transitive closure of the union of the r oriented graphs obtained until round r represents the set of all pairs of elements whose relative order is known at the end of round r.

Suppose we performed r rounds where $r > 0$ is some integer. Consider any function of V that can be computed using the comparisons performed in these r rounds without any further comparisons of elements in V. Our model defines such a function to be *computable following round* r. Note that this definition suppresses all computational steps that do not involve comparisons of elements in V. Which comparisons to perform at round $r + 1$ and the input for each such comparison should be functions which are computable following round r. We are interested in algorithms that sort the elements of V or select an element of a given rank from V.

2.2. A graph theoretic lemma. The following lemma is crucial in all our lower bound proofs.

LEMMA 2.1. *Every graph with n vertices and at most dn edges, $d \geqq 1$ integer, contains an induced subgraph with $\lceil n/4 \rceil$ vertices and maximum degree $< 4d$, which has a $4d$ proper vertex coloring with color classes $V_1, V_2, \cdots, V_{4d}$ such that for each $1 \leqq i, j \leqq 4d$ and each $v \in V_i$, v has at most $2^{|i-j|-1}$ neighbors in V_j.*

Proof. Let $G = (V, E)$ be a graph with n vertices and at most dn edges. Omit successively the highest degree vertex $\lfloor n/2 \rfloor$ times. We are left with an induced subgraph K on a set U of $\lceil n/2 \rceil$ vertices and maximum degree $< 4d$ (otherwise, we omitted at least $\lfloor n/2 \rfloor \cdot 4d \geqq nd$ edges) and with at most $\frac{1}{2}nd$ edges (we omitted at least half of the edges). By a standard result from extremal graph theory, K has a proper $4d$ vertex coloring. Let $U_1, \cdots, U_{4d}$ be the color classes. For every vertex u of K, let $N(u)$ denote the set of all neighbors of u in K. For a permutation π of $1, 2, \cdots, 4d$ and any vertex u of K, define the π-degree $d(\pi, u)$ of u as follows. Let i satisfy $u \in U_{\pi(i)}$; then

$$d(\pi, u) = \sum_{j=1}^{4d} |N(u) \cap U_{\pi(j)}|/2^{|i-j|}.$$

Now consider the sum $\sum_{u \in U} d(\pi, u)$. We claim that the expected value of this sum over all the permutations π of $\{1, \cdots, 4d\}$ is at most $|U|$. Indeed for a random permutation π, the probability that a fixed edge $\{u, v\}$ in U contributes $2 \cdot 1/2^r$, $r > 0$, to the sum ($1/2^r$ to $d(\pi, u)$ and $1/2^r$ to $d(\pi, v)$) is at most $(4d - r)/\binom{4d}{2} \leqq 1/2d$ for all $r > 0$. But there are at most $\frac{1}{2}nd$ edges and therefore this sum is at most $\frac{1}{2}nd \cdot 1/2d \sum_{r>0} 2/2^r \leqq n/2 \leqq |U|$. Hence, there is a fixed permutation σ such that $\sum_{u \in U} d(\sigma, u) \leqq |U|$. It follows that $d(\sigma, u) \leqq 2$ for at least $|U|/2 \geqq n/4$ vertices of K. Let W be a set of $\lceil n/4 \rceil$ of these vertices. Let H be the induced subgraph of G on W, and define $V_i = U_{\sigma(i)} \cap W$ ($1 \leqq i \leqq 4d$). Clearly for every $1 \leqq i, j \leqq 4d$ and every $v \in V_i$, $\sum_{j=1}^{4d} |N(v) \cap V_j|/2^{|i-j|} \leqq 2$ and thus v has at most $2^{|i-j|+1}$ neighbors in V_j. This completes the proof. □

Remark. In the last lemma we can replace $2^{|i-j|+1}$ by $O(f(|i-j|))$ for every function f such that $\sum_{i>0} 1/f(i) < \infty$. For our purpose here the function 2^i suffices.

2.3. Lower bounds for sorting. In this section we prove the lower bounds in Theorems 1.1, 1.2, and 1.3.

LEMMA 2.2. *Every graph $G = (V, E)$ with n vertices and at most $n \cdot d$ edges, where $d = o(n)$ and $d = \Omega(\log n)$, has an acyclic orientation whose transitive closure has at most $\binom{n}{2} - \Omega((n^2/d) \log (n/d))$ edges.*

Proof. By Lemma 2.1 there is a subset W of cardinality $\lfloor n/4 \rfloor$ of V and a proper $4d$ vertex coloring of the induced subgraph of G on W with color classes $V_1, \cdots, V_{4d}$ satisfying the conclusions of the lemma. Put $V_0 = V - W$ and orient each edge (u, v) of G with $u \in V_i$, $v \in V_j$ and $0 \leqq i < j \leqq 4d$ from u to v. The other edges of V (that join two members of V_0) will be oriented in an arbitrary acyclic order. Let T be the transitive closure of this oriented graph. For $v \in V$, let $N_T(v)$ denote the set of neighbors of v in T. Suppose $v \in V_i$, $1 \leqq i < i + j \leqq 4d$. We claim that the number of directed paths in our oriented G that start at v and end at some member of $\bigcup_{r=1}^{j} V_{i+r}$ is at most 2^{3j}. Indeed, each such path must be of the form $v, v_{i_1}, v_{i_2} \cdots v_{i_t}$, where

$$i < i_1 < i_2 < \cdots < i_t \leqq i + j, \qquad v_{i_1} \in V_{i_1}, \cdots, v_{i_t} \in V_{i_t}.$$

There are 2^j possibilities for choosing $i_1, i_2, \cdots, i_t$, and since each vertex of the path is a neighbor of the previous one, there are at most $2^{i_1 - i + 1}$ choices for v_{i_1}, $2^{i_2 - i_1 + 1}$ choices for v_{i_2}, etc. Hence the total number of paths is smaller than 2^{3j} and thus if $v \in V_i$ then

$$\left| N_T(v) \cap \left(\bigcup_{r=1}^{j} V_{i+r} \right) \right| < 2^{3j}.$$

Put $r = \lfloor \frac{1}{6} \log_2 (n/4d) \rfloor$, and partition the set of blocks $V_1, V_2, \cdots, V_{4d}$ into $s = \lceil 4d/r \rceil$ blocks $W_1, \cdots, W_s$ of consecutive $V_i - s$, each containing at most r blocks. By the preceding paragraph, the total number of edges of T in each block W_i is not bigger than $|W_i| \cdot 2^{3r} \leqq |W_i| \sqrt{n/4d}$. Thus there are at least

$$\sum_{i=1}^{s} \binom{|W_i|}{2} - \frac{n}{4} \left(\frac{n}{4d} \right)^{1/2}$$

pairs of elements that are not adjacent in T. By the convexity of the function $g(x) = \binom{x}{2}$

$$\sum_{i=1}^{s} \binom{|W_i|}{2} \geqq s \binom{n/4s}{2} = \Omega\left(\frac{n^2 \log (n/d)}{d} \right)$$

and thus T does not contain at least

$$\Omega\left(\frac{n^2}{d} \log \frac{n}{d} \right) - \frac{n}{4} \left(\frac{n}{4d} \right)^{1/2} = \Omega\left(\frac{n^2}{d} \log \frac{n}{d} \right)$$

edges. This completes the proof. □

We can now prove the lower bounds in Theorems 1.1, 1.2, and 1.3.

To prove the lower bound in Theorem 1.1, consider any two rounds algorithm that sorts a set V of n elements. The first round of the algorithm consists of some set E of comparisons. Define d by $d = \lceil |E|/n \rceil$. Clearly we may assume that $d = o(n^{2/3})$ and $d = \Omega(n^{1/3})$. By Lemma 2.2 the graph $G = (V, E)$ has an acyclic orientation whose transitive closure misses $\Omega((n^2/d) \log n)$ edges. If the answers in the first round correspond to this orientation then clearly in the second round the algorithm has to compare all these $\Omega((n^2/d) \log n)$ pairs. Thus, by the trivial inequality $a + b \geqq 2\sqrt{ab}$

$$c(2, n) \geqq nd + \Omega\left(\frac{n^2}{d} \log n \right)$$

$$\geqq \Omega(n^{3/2} (\log n)^{1/2}), \quad \text{as needed.}$$

The proof of Theorem 1.3(i) is analogous. If a two rounds sorting algorithm uses $c \cdot n \log n$ comparisons in the first round, then by Lemma 2.2, it must use $\Omega(n^2/c)$ comparisons in the second round.

Theorem 1.2 is derived from the lower bound of Theorem 1.1 proved above by induction on k, starting with $k = 2$. For $k = 2$, the result is just the statement of Theorem 1.1. Suppose, by induction, that $c(k, n) \geq c_k n^{1+1/k}(\log n)^{1/k}$, where $c_k > 0$ is a constant, depending only on k. Consider an algorithm for sorting a set V of n elements in $k+1$ rounds. Let E be the set of comparisons between pairs of elements of V made in the first round. As before, E corresponds to a set of edges of a graph $G = (V, E)$. Define d by $|E| = d \cdot n$. By a standard result from extremal graph theory (that follows, e.g., from the trivial part of Lemma 2.1), any graph with m vertices and average degree f, contains an independent set of size $\Omega(m/(1+f))$. By a repeated application of this, we conclude that G contains $\Omega(1+d)$ pairwise disjoint independent sets, each of size $\Omega(n/(1+d))$. Denote these sets by $V_1, \cdots, V_s$ ($s = \Omega(1+d)$) and define $V_0 = V - \bigcup_{i=1}^{s} V_i$. Restrict our attention now only to linear orders on V for which each $v_i \in V_i$ is smaller than each $v_j \in V_j$, for all $0 \leq i < j \leq s$. Clearly, if $0 < i \leq s$, and $u, v \in V_i$ we do not have any information about the relative order of u and v from the results of the first round, and such an information can be obtained only from comparisons between elements of V_i. Thus, in the next k rounds, all the sets $V_1, \cdots, V_s$ have to be sorted. By the induction hypothesis the number of comparisons for this task is at least

$$\sum_{i=1}^{s} c_k |V_i|^{1+1/k}(\log |V_i|)^{1/k} \geq \Omega_k\left((1+d) \cdot \left(\frac{n}{1+d}\right)^{1+1/k} \cdot \left(\log\left(\frac{n}{1+d}\right)\right)^{1/k}\right).$$

The total number of comparisons is thus at least

$$nd + \Omega\left(n^{1+1/k}\left(\log \frac{n}{1+d}\right)^{1/k} \Big/ (1+d)^{1/k}\right).$$

We can easily check that this number is $\geq \Omega(n^{1+1/(k+1)} \cdot (\log n)^{1/(k+1)})$. (Indeed, at least one of the two summands must be that big.) This completes the induction and Theorem 1.2 follows. □

2.4. Lower bounds for selecting. In this section we prove the lower bound in Theorem 1.4.

LEMMA 2.3. *Any algorithm for finding the median in two rounds that uses at most nd comparisons in the first round where $\frac{1}{6} \log n < d = o(n)$ must use in the worst case $\Omega(n^2 \log^2 (n/d)/d^2)$ comparisons in the second round.*

Proof. As usual we look at the elements as vertices and the comparisons as edges. Then we look at $G = (V, E)$, which is the first round graph. By Lemma 2.1 there is a subset W, $|W| = \lceil n/4 \rceil$ of V and a proper $4d$ vertex coloring of the induced subgraph G on W with color classes $V_1, \cdots, V_{4d}$, satisfying the conclusions of the lemma. Let $|V_i| = x_i$, $r = \lfloor \frac{1}{6} \log_2 (n/(4d)) \rfloor \leq d$. We claim that there is a t, $0 \leq t \leq 4d - r$ and r successive classes $V_{t+1}, \cdots, V_{t+r}$ such that $l_t = \sum_{i=t+1}^{t+r} x_i \geq nr/20d$. The claim follows from

$$l_0 + l_r + l_{2r} + \cdots + l_{r(\lfloor 4d/r \rfloor - 1)} + (l_{4d-r}) \geq \sum_{i=1}^{4d} x_i \geq \frac{n}{4}.$$

In this sum there are $\lfloor 4d/r \rfloor + 1$ elements; therefore there must be t and l such that $l = l_t \geq (n/4)/(4d/r + 1) \geq nr/(4(4d+r)) \geq nr/(4 \cdot 5d) = nr/20d$ as claimed.

Take an arbitrary set of vertices of size $\lfloor n/2 - (\sum_{i=1}^{t} x_i + l/2) \rfloor$ of $V' = V - W$ and define it as V_0. (This is possible since $|V_0| \leq \lfloor n/2 \rfloor \leq \lfloor 3n/4 \rfloor = |V'|$.) Define $V_{4d+1} = V' - V_0$. Now orient each edge (u, v) of G with $u \in V_i$, $v \in V_j$ for $0 \leq i < j \leq 4d+1$ from u to v (i.e., u is smaller than v).

The other edges of V (that join two members of V_0 or two elements of V_{4d+1}) will be oriented in an arbitrary acyclic order. Put

$$A = \bigcup_{i=0}^{t} V_i, \quad B = \bigcup_{i=t+1}^{t+r} V_i, \quad C = \bigcup_{i=t+r+1}^{4d+1} V_i,$$

$$|A| = \left\lfloor \frac{n-l}{2} \right\rfloor, \quad |B| = l, \quad |C| = \left\lceil \frac{n-l}{2} \right\rceil.$$

We now let the algorithm know for free the total order of A and C and that for every $v_a \in A$, $v_b \in B$, $v_c \in C$, $v_a < v_b < v_c$.

We now prove that for each $u \in B$ there are less than $l/4$ elements of B that are known to be smaller than u and less than $l/4$ elements of B that are known to be greater than u. To this end let T be the transitive closure of the above oriented graph. For $v \in V$, let $N_T(v)$ denote the set of neighbors of v in T. Suppose that $v \in V_i$, $t+1 \leq i < i+j \leq t+r$. We claim that the number of directed paths in our oriented G that start at v and end at some member of $\bigcup_{s=1}^{j} V_{i+s}$ is at most 2^{3j}. Indeed, each such path must be of the form $v_{i_1}, \cdots, v_{i_s}$, where $v_{i_1} \in V_{i_1}, \cdots, v_{i_s} \in V_{i_s}$. There are 2^j possibilities for choosing $i_1, i_2, \cdots, i_s$ and since each vertex of the path is a neighbor of the previous one there are at most $2^{i_1 - i + 1}$ choices for v_{i_1}, etc. Hence the total number of paths is smaller than 2^{3j} and thus if $v \in V_i$ then $|N_T(v) \cap (\bigcup_{s=1}^{j} V_{i+s})| < 2^{3j}$. As $t+1 \leq i < i+j \leq t+r$, $j < r$ we conclude that the number of elements of B known to be less than v is less than $2^{3r} \leq 2^{(1/2)\log_2(n/4d)} = \sqrt{n/4d} < \frac{1}{4}nr/20d = \frac{1}{4}l$. (In the last inequality we assume that n is large enough, and that $d = o(n)$.) The number of elements of B known to be greater than v can be bounded analogously. Note that the condition for two uncompared elements to possibly be a median and an element next to the median is that together they have less than $n/2$ elements known to be less than at least one of them, and less than $n/2$ elements known to be greater than at least one of them. Therefore, we must compare each pair of such elements in the second round, since, otherwise, even knowing all the order apart from the relative order of these two might still leave each of them a possible candidate for the median. Suppose $u, v \in B$. There are less than $2 \cdot l/4 = l/2$ elements of B and $|A|$ elements of A (which sum up to be less than $l/2 + |A| = n/2$ elements) that are known to be smaller than at least one of u on v. Similarly there are less than $n/2$ elements that are known to be greater than at least one of them. If u and v are not adjacent in T, they satisfy the previous condition and therefore must be compared in the second round.

For each element $v \in B$, there are at least $l - 2 \cdot l/4 = l/2$ elements of B that are not adjacent to it in T. Altogether, there are l elements in B, therefore there are at least $\frac{1}{2} \cdot l \cdot l/2 = l^2/4$ comparisons we must perform in the second round, which means $\Omega(n^2 r^2/d^2) = \Omega(n^2 \log^2 (n/d)/d^2)$. This completes the proof. □

We now prove the lower bound of Theorem 1.4 for $k = 2$.

PROPOSITION 2.4. $s(2, n) = \Omega(n^{4/3} \log^{2/3} n)$

Proof. Consider any two rounds algorithm that finds the median of a set V of n elements. The first round consists of some set E of comparisons. If

$$|E| = \Omega(n^{4/3} \log^{2/3} n),$$

there is nothing to prove. Therefore $|E| = O(n^{4/3} \log^{2/3} n)$. Define $d = \lceil |E|/n \rceil$. By the previous lemma we need $\Omega(n^2 \log^2 (n/d)/d^2) = \Omega(n^2 \log^2 n/(n^{1/3} \log^{2/3} n)^2) = \Omega(n^{4/3} \log^{2/3} n)$ comparisons in the second round, which completes the proof. □

Next we prove the lower bound in Theorem 1.4 by induction on k. For $k = 2$ this is just the previous proposition. Consider an algorithm for finding the median of a set V of n elements in $k+1$ rounds. Let E be the set of comparisons between pairs of elements of V made in the first round. As before, E corresponds to a set of edges of a graph $G = (V, E)$. Define $d = \lceil |E|/n \rceil$. By a standard result from extremal graph theory, any graph with n vertices and average degree $\leq 2d$ contains an independent set V' of size $l = \Omega(n/(1+2d))$. Now we restrict our attention to linear orders on V for which the set of ranks of V' is $[(n-l)/2, (n+l)/2]$, and all the other elements are known to be greater or smaller than each element of V'. Clearly we cannot get any information on the order in V' and the median by comparisons including elements out of V'. In V' all the orders are possible and we should find the median there in the next k rounds. Therefore by the induction hypothesis,

$$S(k+1, n) \geq n(d-1) + s(k, l) = n(d-1) + \Omega(l^{1+1/(2^k-1)} \log^{2/(2^k-1)} l)$$
$$= (d-1)n + \Omega\left(\left(\frac{n}{(1+2d)}\right)^{1+1/(2^k-1)} \log^{2/(2^k-1)}\left(\frac{n}{1+2d}\right)\right)$$
$$\geq \Omega(n^{1+1/(2^{k+1}-1)} \log^{2/(2^{k+1}-1)} n).$$

(Indeed, at least one of the last two summands must be at least that big.) This completes the induction and the proof. □

3. The upper bounds. In this section we prove the upper bounds in Theorems 1.1 and 1.3. The proof of the upper bounds in Theorem 1.4 is similar, and is omitted. In some of the probabilistic lemmas proven below we can simplify the computations a little by applying Chernoff's inequality (cf., e.g., [ES74]). As this simplification is not essential we prefer to use only the direct estimate $\binom{a}{b} \leq (ea/b)^b$. We start with a few lemmas. The first one (which is not essential but somewhat simplifies the proof) is the following result, which is proved in [AKSS86a] (see also [Pi87]).

LEMMA 3.1. *Let G be a graph in which any two disjoint sets of vertices are joined by an edge. Then for every set X containing at least $5a$ vertices, there exists a set Y disjoint from X and containing at most a vertices such that every set Z disjoint from $X \cup Y$ of $z \leq a$ vertices has at least $2z$ neighbors in X.* □

Our next lemma deals with random graphs. Let $G = (V, E)$ be a random graph on n vertices in which each edge is independently present with probability p, where $p = (100 \log n)/a$ and $a > 100 \log n$ will be chosen later. In what follows we always assume that n is sufficiently large, and when we say that G has a property P "almost surely" we mean that the probability that G does not satisfy P tends to zero as n tends to infinity.

LEMMA 3.2. *G satisfies the following almost surely:*

(i) *The number of edges of G is at most $200n^2 \log n/a$.*

(ii) *There are no two disjoint sets A, $B \subseteq V$, with $|A| = |B| = a$ so that each $b \in B$ has less than $\log n$ neighbors in A.*

(iii) *There are no two disjoint sets X, $Y \subseteq V$, with $|X| = x \leq a/e^{\sqrt{\log n}}$ and $|Y| = \lfloor x \log^{1/4} n \rfloor$ such that each $v \in X$ has at least $\log n$ neighbors in Y.*

Proof. Part (i) is trivial and follows, e.g., from Chebyshev's inequality and the fact that the mean and the variance of the number of edges in G are $\binom{n}{2}p < 100n^2 \log n/a$ and $\binom{n}{2}p(1-p) < 100n^2 \log n/a$, respectively. (It also follows, of course, from the standard estimates of binomial distributions.) To prove (ii), fix a subset A of cardinality a of V and let $b \notin A$ be a vertex. Let E_b denote the event that b has less than $\log n$ neighbors in A. Clearly Prob $(E_b) = \sum_{0 \leq i < \log n} b(a, i, p)$ where $b(a, i, p) = \binom{a}{i} p^i (1-p)^{a-i}$. We can easily check that for all $0 \leq i < \log n$, $b(a, i+1, p) > 2b(a, i, p)$, and hence

$$\text{Prob}(E_b) \leq 2b(a, \lfloor \log n \rfloor, p) \leq 2\left(\frac{ea}{\log n}\right)^{\log n} \cdot \left(\frac{100 \log n}{a}\right)^{\log n} \left(1 - \frac{100 \log n}{a}\right)^{a - \log n}$$
$$\leq 2 \cdot (100e)^{\log n} \cdot e^{-100 \log n + 100 \log^2 n/a} < \frac{1}{n^{10}},$$

where the last inequality holds for all sufficiently large n, taking into account the fact that $a > 100 \log n$. It follows that the probability that there are A, $B \subseteq V$ with $|A| = |B| = a$ so that each $b \in B$ has less than $\log n$ neighbors in A is at most

$$\binom{n}{a}\binom{n}{a} \cdot \frac{1}{n^{10a}} < \frac{1}{n^{8a}} \to 0 \quad \text{as } n \to \infty.$$

This completes the proof of (ii).

The probability that G contains two sets X and Y as in part (iii) is at most

$$\binom{n}{x}\binom{n}{x \log^{1/4} n} \cdot \left[\binom{x \log^{1/4} n}{\log n} \cdot \left(\frac{100 \log n}{a}\right)^{\log n}\right]^x.$$

Indeed the first two binomial coefficients bound the number of possible sets X and Y, respectively, and the last factor bounds the probability that each $v \in X$ has at least $\log n$ neighbors in Y. This probability is bounded by

$$n^{2x \log^{1/4} n} (ex)^{x \log n} \cdot \left(\frac{100 \log n}{a}\right)^{x \log n} \leq n^{2x \log^{1/4} n} \left(\frac{100e \log n}{e^{\log^{1/2} n}}\right)^{x \log n} \to 0$$

as n tends to infinity. This completes the proof of the lemma. □

The next lemma is similar though somewhat more complicated than the key lemma in [Pi87].

LEMMA 3.3. *If n elements are compared according to the edges of a graph G that satisfies all the assertions of Lemma 3.2 then, for every rank m, all but at most $O(a \log n/(\log \log n))$ elements will be known to be too small or too large to have this rank.*

Proof. Put $b = \lceil \sqrt{\log n} \log_2 c \rceil$ and $c = \lceil 4 \log n/\log \log n \rceil$. We show that after all the comparisons all but at most $a(1 + 6b + 2c) = O(a \log n/\log \log n)$ of the elements with rank greater than m will be known to have rank greater than m. Combining this with an analogous argument for the elements with rank less than m we get the desired result.

If $m > n - a(1 + 6b + 2c)$ there is nothing to prove. Otherwise, let C_0 contain the smallest m elements, let C_1 contain the next $6a \cdot b$ elements, let C_2 contain the next $2a \cdot c$ elements, and let C_3 contain the remaining elements. Since $|C_1 \cup C_2| = 6ab + 2ac$ it suffices to prove that C_3 contains at most a elements whose rank is not known to be greater than m. Let $C_1 = C_{1,1} \cup C_{1,2} \cdots \cup C_{1,b}$ be a partition of C_1 into b pairwise disjoint sets of $6a$ consecutive elements each, where C_{11} is the set of smallest $6a$ elements of C_1, C_{12} the next $6a$ elements of C_1, etc. Similarly, let $C_2 = C_{2,1} \cup \cdots \cup C_{2,c}$ be a partition of C_2 into c pairwise disjoint sets of $2a$ consecutive elements each, where C_{21} is the set of smallest $2a$ elements of C_2, $C_{2,2}$ the next $2a$ elements of C_2, etc. We now classify the elements in $C_1 \cup C_2$ as "good" or "bad" as follows. All the elements of C_{11} are good. Suppose $1 \leq i < b$ and all the elements of $C_{1,i}$ have already been classified, and that there are at least $5a$ good elements in $C_{1,i}$. To classify the elements of $C_{1,i+1}$, apply Lemma

3.1 with X being the set of good elements in $C_{1,i}$. (Notice that by Lemma 3.2(ii), G satisfies the assumptions of Lemma 3.1.) Classify as bad those elements of $C_{1,i+1}$ that belong to the resulting set Y, whose size is at most a. The other elements of $C_{1,i+1}$, whose number is at least $5a$, are good. This completes the classification of elements in C_1. Now put $C_{2,0} = C_{1,b}$. Suppose $0 \leq i < c$ and suppose all elements of $C_{2,i}$ have already been classified and that there are at least a good elements in $C_{2,i}$. An element of $C_{2,i+1}$ is good if it has at least $\log n$ good neighbors in $C_{2,i}$. Otherwise, it is bad. Since G satisfies all the assertions of Lemma 3.2 we conclude, by (ii), that there are at least a good elements in $C_{2,i+1}$. Moreover, by Lemma 3.2(iii), any set X of $x \leq a/e^{\sqrt{\log n}}$ good elements in $C_{2,i+1}$ has at least $x \log^{1/4} n$ good neighbors in $C_{2,i}$. Similarly, by Lemma 3.1, any set Z of $z \leq a$ good elements in $C_{1,i+1}$ has at least $2z$ good neighbors in $C_{1,i}$.

Since in G there is an edge between any two disjoint sets of size a, and since there are at least a good elements in $C_{2,c}$, it follows that all but at most a elements in C_3 have at least one good neighbor in $C_{2,c}$. Let $v \in C_3$ be such an element, and let $\phi \neq X$ be the set of its good neighbors in $C_{2,c}$. Clearly, v is known to be larger than all members of X. If $|X| < a/e^{\sqrt{\log n}}$ it has at least $|X| \cdot \log^{1/4} n$ good neighbors in $C_{2,c-1}$, and by transitivity, v is known to be larger than these elements as well. Continuing in this manner, we conclude that v is known to be larger than at least $a/e^{\sqrt{\log n}}$ good elements of $C_{2,0} = C_{1,b}$. Let Z be the set of these elements. If $|Z| \leq a$, Z has at least $2|Z|$ good neighbors in $C_{1,b-1}$. Continuing in this manner, we conclude that v is known to be larger than at least a elements in $C_{1,1}$, and hence it is known to be larger than all but at most a elements in C_0. Consequently, the rank of v is known to be greater than m. This discussion was true for all but at most a elements of C_3. Hence, there are at most $a + |C_1 \cup C_2|$ elements whose rank is greater than m, that are not known to have rank greater than m. This completes the proof of the lemma. □

COROLLARY 3.4. *If n elements are compared according to the edges of a graph G as in Lemma 3.2, then for every interval $[i, j]$ of $O(a \log n/\log \log n)$ successive ranks there are only $O(a \log n/\log \log n)$ elements whose ranks can possibly belong to this interval. Hence, we can form $O(n \log \log n/a \log n)$ (not necessarily disjoint) subsets of elements of size $O(a \log n/\log \log n)$ each, so that the total ordering will be known by sorting each of these sets separately.*

Proof. By the previous lemma, all but $O(a \log n/\log \log n)$ of the elements with ranks less than i are known to have ranks less than i, and all but $O(a \log n/\log \log n)$ of the elements with ranks more than j are known to have ranks more than j. As only $O(a \log n/\log \log n)$ elements have ranks in the interval $[i, j]$ there are only $O(a \log n/\log \log n)$ candidates for the interval. By partitioning all ranks into $t = O(n \log \log n/a \log n)$ contiguous intervals of size $O(a \log n/\log \log n)$ each we split the sorting problem into t smaller subproblems, as needed. □

Proof of the Upper Bound of Theorem 1.1. Let G be a graph as in Lemma 3.2 with $a = \lfloor n^{1/2}(\log \log n)^{1/2} \rfloor$. In the first round of the algorithm compare the elements according to G. This gives $O(n^2 \log n/a)$ comparisons. By the last corollary, the number of comparisons left for the second round is at most $O(n \log \log n/a \log n \cdot (a \log n/\log \log n)^2)$. Altogether we get

$$c(2, n) \leq O(n^2 \log n/a + na \log n/\log \log n) = O(n^{3/2} \log n/(\log \log n)^{1/2}). \qquad \square$$

The proof of Theorem 1.3(ii) is similar to that of Theorem 1.1. Instead of Lemma 3.2 we need a similar lemma for random graphs. Define $a = \lfloor n/\log n \rfloor$ and $p = 100 \log n \log \log n/n$. Let $G = (V, E)$ be a random graph on n vertices in which each edge is independently present with probability p. Again, we always assume that n is sufficiently large.

LEMMA 3.5. *G satisfies the following almost surely:*

(i) *The number of edges of G is at most* $200n \log n \log \log n$.

(ii) *There are no two disjoint sets $A, B \subseteq V$, with $|A| = |B| = a$, so that each $b \in B$ has less than* $\log \log n$ *neighbors in A.*

(iii) *There are no two disjoint sets $X, Y \subseteq V$, with $|X| = x = a/r \leq a/\log^2 n$ and $|Y| = \lfloor x(\log \log n)^{1/2} \rfloor$ such that each $v \in X$ has at least* $\log \log n$ *neighbors in Y.*

Proof. Part (i) is trivial and follows, e.g., from Chebyshev's inequality. To prove (ii), fix $A \subseteq V$, $|A| = a$, and $b \in V \setminus A$ and let E_b denote the event that b has less than $\log \log n$ neighbors in A. Clearly $\text{Prob}(E_b) = \sum_{0 \leq i < \log \log n} b(a, i, p)$ where $b(a, i, p) = \binom{a}{i} p^i (1-p)^{a-i}$. Since $b(a, i+1, p) > 2b(a, i, p)$, for all $o \leq i < \log \log n$,

$$\text{Prob}(E_b) \leq 2b(a, \lfloor \log \log n \rfloor, p) \leq 2 \cdot \left(\frac{ea}{\log \log n}\right)^{\log \log n} p^{\log \log n}(1-p)^{a - \log \log n} < e^{-50 \log \log n}.$$

Therefore, the probability that there are $A, B \subseteq V$ as in (ii) is at most

$$\binom{n}{a}\binom{n}{a} \cdot e^{-50a \log \log n} \leq (e \log n)^{2a} \cdot e^{-50a \log \log n} \to 0$$

as $n \to \infty$. This establishes (ii).

The probability that G contains two sets X and Y as in (iii) can be bounded by

$$\binom{n}{x}\binom{n}{x(\log \log n)^{1/2}} \cdot \left[\binom{x(\log \log n)^{1/2}}{\log \log n} \cdot p^{\log \log n}\right]^x.$$

Here the first two binomial coefficients bound the number of possible sets X and Y, respectively, and the last factor bounds the probability that each $v \in X$ has at least $\log \log n$ neighbors in Y. This probability is at most

$$\left(\frac{er \log n}{(\log \log n)^{1/2}}\right)^{2n(\log \log n)^{1/2}/r \log n} \cdot \left(\frac{en}{r \log n (\log \log n)^{1/2}} \cdot \frac{100 \log n \log \log n}{n}\right)^{n \log \log n/r \log n},$$

which tends to zero as n tends to infinity, since $r > \log^2 n$. This completes the proof. □

The proof of the next lemma is analogous to that of Lemma 3.3 (simply replace Lemma 3.2 by Lemma 3.5, replace b by $b' = \lceil \log_2 \log^2 n \rceil$, and c by

$$c' = \lceil 2 \log n \log \log \log n \rceil).$$

We omit the details.

LEMMA 3.6. *If n elements are compared according to the edges of a graph G that satisfies all the assertions of Lemma 3.5 then, for every rank m, all but at most $O(a \log n/\log \log \log n) = O(n/\log \log \log n)$ elements will be known to be too small or too large to have the rank.*

Theorem 1.3(ii) follows from the last lemma. As in Corollary 3.4 we observe that if n elements are compared according to the $O(n \log n \log \log n)$ edges of a graph G as in Lemma 3.5, then for every interval of $O(n/\log \log \log n)$ successive ranks there are only $O(n/\log \log \log n)$ possible candidates. It then follows, as in Corollary 3.4, that at most $O(n^2/\log \log \log n) = o(n^2)$ pairs can possibly remain unknown.

REFERENCES

[AA87] N. ALON AND Y. AZAR, *The average complexity of deterministic and randomized parallel comparison sorting algorithms*, in Proc. 28th Annual IEEE Foundations of Computer Science, Los Angeles, CA, 1987, pp. 489–498; also: SIAM J Comput., to appear.

[AA88] ———, *Finding an approximate maximum*, SIAM J. Comput., to appear.

[AAV86] N. ALON, Y. AZAR, AND U. VISHKIN, *Tight complexity bounds for parallel comparison sorting*, in Proc. 27th Annual IEEE Foundations of Computer Science, Toronto, Ontario, Canada, 1986, pp. 502–510.

[AKSS86a] M. AJTAI ET AL., *Deterministic selection in O(log log n) parallel time*, in Proc. 18th Annual ACM Symposium on Theory of Computing, Berkeley, CA, 1986, pp. 188–195.

[AKSS86b] ———, *Almost sorting in one round*, Adv. Comput. Res., to appear.

[Ak85] S. AKL, *Parallel Sorting Algorithms*, Academic Press, New York, 1985.

[Al85] N. ALON, *Expanders, sorting in rounds and superconcentrators of limited depth*, in Proc. 17th Annual ACM Symposium on Theory of Computing, Providence, Rhode Island, 1985, pp. 98–102.

[AV87] Y. AZAR AND U. VISHKIN, *Tight comparison bounds on the complexity of parallel sorting*, SIAM J. Comput., 3 (1987), pp. 458–464.

[BB87] B. BOLLOBÁS AND G. BRIGHTWELL, *Graphs whose every transitive orientation contains almost every relation*, Israel J. Math., 59 (1987), pp. 112–128.

[BHe85] B. BOLLOBÁS AND P. HELL, *Sorting and graphs*, in Graphs and Orders, I. Rival, ed., D. Reidel, Boston, MA, 1985, pp. 169–184.

[Bo86] B. BOLLOBÁS, *Random Graphs*, Academic Press, New York, 1986, Chap. 15.

[BHo82] A. BORODIN AND J. E. HOPCROFT, *Routing, merging and sorting on parallel models of computation*, in Proc. 14th Annual ACM Symposium on Theory of Computing, San Francisco, 1982, pp. 338–344.

[BR81] B. BOLLOBÁS AND M. ROSENFELD, *Sorting in one round*, Israel J. Math., 38 (1981), pp. 154–160.

[BT83] B. BOLLOBÁS AND A. THOMASON, *Parallel sorting*, Discrete Appl. Math., 6 (1983), pp. 1–11.

[ES74] P. ERDOS AND J. SPENCER, *Probabilistic Methods in Combinatorics*, Academic Press, New York, 1974, p. 18.

[HH80] R. HÄGGKVIST AND P. HELL, *Graphs and parallel comparison algorithms*, Congress Numer., 29 (1980), pp. 497–509.

[HH81] ———, *Parallel sorting with constant time for comparisons*, SIAM J. Comput., 10 (1981), pp. 465–472.

[Kn73] D. E. KNUTH, *The Art of Computer Programming, Vol. 3: Sorting and Searching*, Addison Wesley, Reading, MA, 1973.

[Pi85] N. PIPPENGER, *Explicit construction of highly expanding graphs*, preprint (1985).

[Pi87] ———, *Sorting and selecting in rounds*, SIAM J. Comput. 16 (1987), pp. 1032–1038.

[Va75] L. G. VALIANT, *Parallelism in comparison problems*, SIAM J. Comput., 4 (1975), pp. 348–355.