SIAM J. COMPUT.
Vol. 17, No. 6, December 1988

THE AVERAGE COMPLEXITY OF DETERMINISTIC AND RANDOMIZED PARALLEL COMPARISON-SORTING ALGORITHMS*

N. ALON†‡ AND Y. AZAR†

Abstract. In practice, the average time of (deterministic or randomized) sorting algorithms seems to be more relevant than the worst-case time of deterministic algorithms. Still, the many known complexity bounds for parallel comparison sorting include no nontrivial lower bounds for the average time required to sort by comparisons n elements with p processors (via deterministic or randomized algorithms). We show that for $p \geqq n$ this time is $\Theta(\log n/\log(1+p/n))$ (it is easy to show that for $p \leqq n$ the time is $\Theta(n \log n/p) = \Theta(\log n/(p/n))$. Therefore even the average-case behaviour of randomized algorithms is not more efficient than the worst-case behaviour of deterministic ones.

Key words. parallel sorting, comparison algorithms, randomized sorting

AMS(MOS) subject classification. 68E05

1. Introduction. Sorting is one of the central problems in computer science. For extensive lists of publications dealing with serial and parallel sorting algorithms see, e.g., [Ak85], [BHe85], [Kn73], and [Th83].

Most of the fastest serial and parallel sorting 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.

It is well known that $\Theta(n \log n)$ binary comparisons are both necessary and sufficient for sorting n elements in the serial comparison tree model. The situation is somewhat more complicated for parallel algorithms. The common parallel comparison model here is the one introduced by Valiant [Va75] (see also [BHo85]), 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 any lower bound in this model implies the same bound for all algorithms, based on comparisons, in any parallel random access machine (PRAM), including PRAMs that allow simultaneous access to the same common memory location for read and write purposes.

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.

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).

Upper and lower bounds for $c(k, n)$ appear in [AA87], [AAV86], [AKS83a], [AKS83b], [AV87], [BT83], [BHe85], [Bo86], [BHo85], [HH80], [HH81], [Pi87]. The

* Received by the editors June 15, 1987; accepted for publication (in revised form) January 14, 1988.

† Department of Mathematics, Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv, Israel.

‡ The research of this author was supported in part by an Allon Fellowship, a grant from the Bat Sheva de Rothschild Foundation, and by the Fund for Basic Research administered by the Israel Academy of Sciences.

best-known bounds for fixed k are (see [AA87], [AAV86], [BHe85])

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

and for general $k \leqq \log n$ (see [AV87], [AAV86])

$$\Omega(kn^{1+1/k}) \leqq c(k, n) \leqq kn^{1+b/k} \quad (1.1)$$

where $b > 0$ is a constant. These bounds imply that the time required for sorting n elements, if p comparisons are performed in each time unit, is $\Theta(\log n/(p/n))$ for $p \leqq n$ and $\Theta(\log n/\log(1+p/n))$ for $p \geqq n$.

These results determine up to a constant factor the (worst-case) time complexity of sorting in the parallel comparison tree model. However, the problem of estimating the *average* (over all orders) running time of the best sorting algorithm, as well as that of determining the time complexity of the best randomized sorting algorithm, is far from being settled. In fact, besides the relatively easy $\Omega(n \log n)$ lower bound for randomized (or average) serial sorting (see, e.g., [AHU74]), there are no known lower bounds for the worst-case or average-time complexity of randomized sorting algorithms at all. Such bounds appear to be important, since in practical situations we are naturally interested in the average running time, and not necessarily in the worst-case behaviour. Similarly, fast randomized parallel sorting algorithms could be extremely helpful in practice.

Proving lower bounds for the average time of (deterministic or randomized) comparison algorithms appears to be much more complicated than obtaining lower bounds for the worst-case time of deterministic ones. In fact, there are several known results that show that for various comparison algorithms the average time, as well as the worst-case time of randomized algorithms, differs asymptotically from the worst-case time of their deterministic counterparts. One such result is due to Reischuk [Re81], who gave a randomized comparison parallel algorithm for selection, whose expected running time is bounded by a constant, using n processors. Together with the $\Omega(\log \log n)$ lower bound of [Va75] for finding the maximum among n elements using n processors, we conclude that there exists a randomized algorithm for selection that performs better than any of its deterministic counterparts. The results of [BHe85] on approximate sorting in one round, those of [Rei85] on integer sorting and those of [AAV86] on sorting in a fixed number of rounds supply several other examples of parallel randomized comparison algorithms that perform better than the corresponding best-known deterministic ones. In view of these examples we might expect that randomized parallel sorting algorithms could work asymptotically faster than deterministic ones. Our main result in this paper is that this is not the case. In fact we prove the following theorem.

THEOREM 1.1. *The average time required for sorting n elements in the best randomized algorithm with p processors,* (i.e., *the best algorithm that performs p comparisons in each time unit*), *is* $\Theta(\log n/\log(1+p/n))$ *for* $p \geqq n$ (*and is, easily,* $\Theta(\log n/(p/n))$ *for* $p \leqq n$).

This matches, up to a constant factor, the worst-case running time for the deterministic case for all values of p.

To prove the lower bound we prove the following proposition.

PROPOSITION 1.2. *The average number of comparisons in the best deterministic algorithm that sorts n elements in k rounds is*

$$\Omega(kn^{1+1/k}) \quad \textit{for all } k \leqq \log n.$$

Note that for $k = \Theta(\log n)$ *this coincides with the known bound for the serial case.*

A parallel algorithm is said to achieve average *optimal speedup* if its average running time is proportional to $\text{Seq}(n)/p$, where $\text{Seq}(n)$ is the lower bound on the average serial running time, n is the size of the problem being considered, and p is the number of processors used.

An immediate consequence of Theorem 1.1 is that if the number p of processors is larger than n by an order of magnitude then it is impossible to design an average optimal speedup randomized comparison sorting algorithm. Note that, for $p = O(n)$, there is an optimal speedup (deterministic and therefore randomized or average deterministic) algorithm, given by the [AKS83a], [AKS83b] sorting network. These results enable us to identify asymptotically the parallelism average break point of sorting, which is the minimum average time that can be achieved by an average optimal speedup algorithm. Specifically, $\Theta(\log n)$ is the average break point for sorting n elements, which is the same break point as that of deterministic algorithms (see [AAV86], [AV87]).

Note that for finding the maximum the average break point is better than the worst-case break point. Reference [Va75] proved that $\Theta(\log \log n)$ is the break point for finding the maximum among n elements but [Re81] proved that $\Theta(1)$ is the average break point of that problem.

The proof of Theorem 1.1 differs considerably and is far more complicated than that of the corresponding result for the worst-case time of deterministic algorithms (which, obviously, follows from it). The main difficulty lies in the proof of Proposition 1.2. As we are dealing with the average-case behaviour, we cannot use the traditional adversary way for choosing the unknown order; we need a new method. Since a direct proof seems elusive, we must prove a stronger result, which implies, e.g., that our bound holds for algorithms that allow, in addition to usual comparisons, questions of the form "is rank $(x) \leqq i$?" For the exact statement of the stronger result see § 2. We are unable to prove Proposition 1.2 without proving this stronger assertion.

The derivation of Theorem 1.1 from Proposition 1.2 is much easier than the proof of Proposition 1.2; however, it is not straightforward. It combines certain probabilistic arguments with a well-known observation of Yao [Ya77] and the upper bound in inequality (1.1). This is described in § 3. The final section, 4, contains concluding remarks, together with an application of our method to selection problems.

2. The average number of comparisons in deterministic algorithms.

2.1. The parallel computation model. Let N be a set of n elements taken from a totally ordered domain. The *parallel comparison model of computation* equivalent to the parallel computation tree model of [BHo85] 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 is two elements of N. The output of each comparison is one of the following two: $<$ or $>$. 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. The comparisons performed in round $r+1$ are chosen, of course, according to the results in all previous rounds.

Recall that $c(k, n)$ denotes the minimum *total* worst-case number of comparisons required to sort n elements in k rounds (over all possible algorithms).

Let $r(k, n)$ denote the *average* total number of comparisons, over all orders, required to sort n elements in k rounds, in the best algorithm (that necessarily stops after k rounds). Clearly $r(k, n) \leqq c(k, n)$. Our objective is to prove Proposition 1.2, which supplies a lower bound for $r(k, n)$, where k is possibly a function of n.

2.2. Legal situations. We prove a stronger result that implies Proposition 1.2. We consider the following situation, called a *legal situation*: We have $s+1$ disjoint sets of elements, denoted Z and $Y_1, \cdots, Y_s$, with a set E of edges (comparisons) between them.

The set Z, $|Z|=m$, $0 \leqq m \leqq n$, is a set of m elements such that the rank of each $z \in Z$ in the total n-order is known. $Y_1, \cdots, Y_s$, $|Y_i| = y_i > 0$, $1 \leqq i \leqq s$ are $s \geqq 0$ sets of elements such that, for each i, the set of y_i ranks of the ith set in the total n-order is known, but all the $y_i!$ orders of the elements of the sets are equally likely. The existing edges (comparisons) we have are only for this round and are only between pairs of elements of the same Y_i, for some i, or between an element of Y_i and an element of Z. The answers to these comparisons are known, but depend, of course, on the actual orders inside the sets Y_i. Let e_i be the number of edges in Y_i. Let $\bar{e}_i$ be the number of edges between an element of Y_i and an element outside Y_i, whose relative order does not follow from the known information about the sets of ranks. Denote $f_i = e_i + \frac{1}{2}\bar{e}_i$ and call it the "f-number" of the set Y_i in this situation.

The following facts are worth noting:

(1) The number of edges $|E|$ satisfies

$$|E| \geqq \sum_{i=1}^{s} \left(e_i + \frac{1}{2}\bar{e}_i\right) = \sum_{i=1}^{s} f_i.$$

(2) The number of elements n satisfies

$$n = m + \sum_{i=1}^{s} y_i.$$

(3) The (known) set of ranks of each Y_i is not necessarily a block of y_i consecutive ranks, and there are no edges between an element of Y_i and an element of Y_j for $i \neq j$. Before the next round of comparisons, all the information about the relative order of an element of Y_i and an element of Y_j is a consequence of either the known sets of ranks of Y_i and Y_j or the results of comparisons to elements of Z and transitivity.

(4) If for some set Y_i, $f_i = 0$ then all the $y_i!$ orders in this set are equally likely for each possibility of the results of the comparisons of the present round (and independently of these results).

2.3. The lower bound. Let A denote the above legal situation. Denote by $F(k, A)$ the average, over all the $\prod_{i=1}^{s} (y_i!)$ possible orders, of the "f-number" of comparisons that are needed to sort the n elements in k more rounds, starting from situation A. Here the "f-number" means that a comparison between an element of Y_i and an element of Y_j (i can be equal to j) is counted as one comparison and a comparison between an element of Y_i and an element of Z is counted as $\frac{1}{2}$ a comparison.

Define

$$g_k(y,f)=\begin{cases}1, & f=0,\\ \left(\frac{c}{4}\right)^{1/k}, & 0<\frac{f}{y}\leqq\frac{1}{4},\\ \left(\frac{cf}{y}\right)^{1/k}, & \frac{f}{y}\geqq\frac{1}{4},\end{cases}$$

where c is some positive constant to be chosen later. Define also

$$\varphi_k(y,f)=k\left[\frac{y^{1+1/k}}{cg_k(y,f)}-y\right].$$

Note that φ_k is a monotone nonincreasing function of f. The key inequality is the following.

THEOREM 2.1. *In the above notation*

$$F(k,A)\geqq\sum_{i=1}^{s}\varphi_k(y_i,f_i) \tag{2.1}$$

for every $k\geqq 1$, *and every legal situation* A.

The (rather lengthy) proof of Theorem 2.1 is given below. It is based on moving from one legal situation to another by finding properties that hold for all orders (and not just for a special one chosen by an adversary). Some inequalities for convex functions and some statistical behaviour of random orders are used as well, together with a reduction of nonlegal situations that are created during the algorithm to legal ones.

Proposition 1.2 for $k\leqq\log n/(\log 2c)$ is the special case of Theorem 2.1 in which A is the legal situation with $Z=\phi$, $s=1$, Y_1 is the set of all elements, and $f_1=0$. For $k=\Theta(\log n)$, Proposition 1.2 follows immediately from the serial bound. Another special case of Theorem 2.1 corresponds to algorithms that allow queries of the form "is rank $(x)\leqq i$" besides usual comparisons. Indeed, suppose we have $2n-1$ elements. Let A be the legal situation in which $|Z|=n-1$, the set of ranks of Z is $\{2,4,\cdots,2n-2\}$, $s=1$, the set of ranks of Y_1 is $\{1,3,\cdots,2n-1\}$ and $f_1=0$. Applying Theorem 2.1 to this situation we conclude that $\Omega(kn^{1+1/k})$ comparisons are needed to sort Y_1 in $k(\leqq\log n)$ rounds, where here, clearly, a comparison to an element of Z corresponds to a query of the form "is rank$(x)\leqq i$?"

Proof of Theorem 2.1. We prove the theorem for every fixed $n\geqq 2$ by induction on k and on the parameters of A with $c=256e$. (We make no effort here to obtain the best possible c.) The base case is left to the end. Our induction hypothesis is that the assertion of Theorem 2.1 holds for any k' and any legal situation A' with sets Z', $|Z'|=m'$ and $Y'_1,\cdots,Y'_{s'}$, $|Y'_i|=y'_i>0$, where

$$m'+\sum_{i=1}^{s'}y'_i=n,$$

provided at least one of the following three cases holds:

(a) $k'<k$.

(b) $k'=k$ and $\sum_{i=1}^{s'}y'_i<\sum_{i=1}^{s}y_i$.

(c) $k'=k$, $\sum_{i=1}^{s'}y'_i\leqq\sum_{i=1}^{s}y_i$ and $s<s'$. (Note that always $s,s'\leqq n$.)

We have to prove the theorem for k and A. We consider two possible cases.

Case 1. There is a set Y_i, with $f_i>0$.

Case 2. For all i, $1\leqq i\leqq s$, $f_i=0$, and $k>1$. (The case $k=1$ and $f_i=0$ for all i will be the base case.)

In Case 1 we can assume, without loss of generality, that $f_s > 0$. Denote $Y = Y_s$, $|Y| = y = y_s$, $f = f_s$.

Subcase 1a. $f \geqq (1/c^{k+1})y^2$. In this case $\varphi_k(y, f) \leqq k(y^{1+1/k}/c(cf/y)^{1/k} - y) \leqq 0$. For each order of Y, let the algorithm know this order for free; i.e., let A' be the legal situation obtained from A by replacing Z by $Z \cup Y$. Since $\sum y_i' < \sum y_i$ and $k' = k$ it follows from the induction hypothesis (case (b)) that

$$F(k', A') \geqq \sum_{i=1}^{s'} \varphi_{k'}(y_i', f_i') = \sum_{i=1}^{s-1} \varphi_k(y_i, f_i) \geqq \sum_{i=1}^{s} \varphi_k(y_i, f_i).$$

Averaging over all orders of Y we conclude that $F(k, A) \geqq \sum_{i=1}^{s} \varphi_k(y_i, f_i)$, as needed. (Note that each comparison that is counted as one comparison in $F(k', A')$ is also counted as 1 in $F(k, A)$ and each comparison that is counted as $\frac{1}{2}$ in $F(k', A')$ is counted either as $\frac{1}{2}$ or as 1 in $F(k, A)$.)

Subcase 1b. $0 < f \leqq y/4$. Since $2f = 2e_s + \bar{e}_s$ there are at most $2f$ elements in Y, which are compared to some other elements (in Y or in Z). Let $Y_s' \subseteq Y$ be a set of $2f \leqq y/2$ elements containing all those that are compared to members of $Y \cup Z$. Also define $Y_{s+1}' = Y \setminus Y_s'$. We let the algorithm know for free the set of ranks of Y_s' and Y_{s+1}'. This corresponds to the legal situation A' obtained from A by splitting Y_s into the two sets Y_s' and Y_{s+1}'. For A', $s' = s+1$, $f_s' \leqq f_s$, $f_{s+1}' = 0$. As $s' > s$ we can apply the induction hypothesis (case (c)) to A' and conclude that

$$F(k, A') \geqq \sum_{i=1}^{s+1} \varphi_k(y_i', f_i') \geqq \sum_{i=1}^{s} \varphi_k(y_i, f_i) - \varphi_k(y, f) + \varphi_k(2f, f) + \varphi_k(y - 2f, 0).$$

In Appendix 1 below we show that for $0 < f \leqq y/4$, $\varphi_k(2f, f) + \varphi_k(y - 2f, 0) \geqq \varphi_k(y, f)$. Thus, for each A' that arises as above $F(k, A') \geqq \sum_{i=1}^{s} \varphi_k(y_i, f_i)$. Averaging over all possible assignments of the two sets of ranks to the elements in Y_s' and in Y_{s+1}', we conclude that $F(k, A)$, which is at least this average, is at least $\sum_{i=1}^{s} \varphi_k(y_i, f_i)$, as needed.

Subcase 1c. (The main case.) $y/4 < f < y^2/c^{k+1}$. Define $t = \lceil 4f/y \rceil (> 1)$, $\beta = \lfloor y/t \rfloor$. Note that $2 \leqq t \leqq 8f/y$, and $y/t \geqq y^2/8f > c^{k+1}/8$, and hence $\beta \geqq 1$ and

$$\beta = \left\lfloor \frac{y}{t} \right\rfloor \geqq \frac{y}{t}\left(1 - \frac{8}{c^{k+1}}\right). \tag{2.2}$$

Randomly partition Y into $t+1$ blocks $B_1, \cdots, B_t, C$ where $|B_i| = \beta$, and $|C| = y - t\beta$ (possibly $|C| = 0$). For each such partition choose a random permutation π of $[1, 2, \cdots, t]$. Assume that each element of C is greater than each element of $Y \setminus C$, and that each element of B_i is greater than each element of B_j if and only if $\pi(i) > \pi(j)$. These choices, together with the assumption that all the orders inside each block are equally likely, give each of the possible $y!$ orders of y with equal probability. For each choice of B_i, C, and π, there are t pairwise disjoint sets $Z_1, \cdots, Z_t$ whose ranges of ranks lie inside those of $B_1, \cdots, B_t$. Let g_i' be the number of comparisons between two elements of B_i, and let g_i'' be the number of comparisons between an element of B_i and an element of Z_i. Also define $g_i = g_i' + \frac{1}{2}g_i''$. Note that g_i', g_i'', and g_i are random variables (whose values depend on the choice of the partition and of π), that $2g_i$ is an integer, and that g_i is the f-number of B_i (since the result of each comparison between members of B_i and members of $N \setminus (B_i \cup Z_i)$ follows from the assumptions on the sets of ranks of each B_i). If $2g_i < \beta$, let G_i be a subset of B_i, $|G_i| = 2g_i$ that contains every member of B_i, which is compared to a member of $B_i \cup Z_i$. (Clearly there are at most $2g_i$ such elements.) If $2g_i \geqq \beta$, define $G_i = B_i$. Also define $N_i = B_i \setminus G_i$. We let our algorithm know for free the sets of ranks of each G_i and each N_i. By the definition of G_i there are no comparisons between G_i and N_i. Hence we now have, for each choice of B_i, C, and π, a new legal situation A', which is obtained from A

by replacing Z by $Z \cup C$ and by replacing $Y = Y_s$ by all the nonempty G_i and N_i (of which there are at least $t \geqq 2$). Note that for A', $k' = k$, $\sum y'_i \leqq \sum y_i$, and $s' > s$. Note also that the f-number of N_i, $f_{N_i} = 0$ for each nonempty N_i, $1 \leqq i \leqq t$. Let $h_i = f_{G_i}$ be the f-number of G_i. Note that $h_i \leqq g_i$. By applying case (c) of the induction hypothesis to A' (and by the fact that $\varphi_k(0,0) = 0$) it suffices to prove that

$$\psi = \psi_1 - \psi_2 \geqq 0 \quad \text{where}$$

$$\psi_1 = E\left(\sum_{i=1}^{t} \varphi_k(|N_i|, 0) + \sum_{i=1}^{t} \varphi_k(|G_i|, h_i)\right), \qquad \psi_2 = \varphi_k(y, f), \tag{2.3}$$

and where the expected value is over all choices of the partition B_i, C, and the permutation π. By the symmetry of the sets B_i

$$E\left(\sum_{i=1}^{t} \varphi_k(|N_i|, 0) + \sum_{i=1}^{t} \varphi_k(|G_i|, h_i)\right) = tE(\varphi_k(|N_1|, 0) + \varphi_k(|G_1|, h_1))$$
$$\geqq tE(\varphi_k(|N_1|, 0) + \varphi_k(|G_1|, g_1)).$$

Denote $g = g_1$. Let A_1 be the event $2g < \beta$ and let $p = p(A_1)$ be its probability. Put $\bar{p} = 1 - p$. In this notation the right-hand side of the last inequality is simply

$$tpE_{A_1}(\varphi_k(\beta - 2g, 0) + \varphi_k(2g, g)) + t\bar{p}E_{\bar{A}_1}(\varphi_k(\beta, g)),$$

where E_{A_1}, $E_{\bar{A}_1}$ are the expectations given A_1, $\bar{A}_1$, respectively. Hence

$$\psi_1 \geqq tpE_{A_1}\left(k\left[\frac{(\beta-2g)^{1+1/k}}{c} - (\beta - 2g)\right] + k\left[\frac{(2g)^{1+1/k}}{c(c/2)^{1/k}} - 2g\right]\right)$$
$$+ t\bar{p}E_{\bar{A}_1}\left(k\left[\frac{\beta^{1+1/k}}{c(cg/\beta)^{1/k}} - \beta\right]\right)$$
$$= \frac{tk}{c}\left[pE_{A_1}\left((\beta-2g)^{1+1/k} + \frac{(2g)^{1+1/k}}{(c/2)^{1/k}}\right) + \bar{p}E_{\bar{A}_1}\frac{\beta^{1+1/k}}{(cg/\beta)^{1/k}}\right] - kt\beta(p+\bar{p}).$$

Put $E_{A_1}(2g) = a_1$, $E_{\bar{A}_1}(2g) = a_2$, $E(2g) = a$. Note that $a_2 \geqq \beta$, $pa_1 + \bar{p}a_2 = a$, $p + \bar{p} = 1$, and $\beta t \leqq y$. By the convexity of the functions $x^{1+1/k}$ and $1/x^{1/k}$ we can apply Jensen's inequality (see, e.g., [HLP59]) to get

$$\psi_1 \geqq \frac{kt}{c}\left[p\left((\beta - a_1)^{1+1/k} + \frac{a_1^{1+1/k}}{(c/2)^{1/k}}\right) + \bar{p}\frac{\beta^{1+1/k}}{(\frac{1}{2}ca_2/\beta)^{1/k}}\right] - ky.$$

Put $\alpha_1 = a_1/\beta$, $\alpha_2 = a_2/\beta$, $\alpha = a/\beta$. Clearly

$$\alpha_2 \geqq 1, \qquad p\alpha_1 + \bar{p}\alpha_2 = \alpha, \tag{2.4}$$

and

$$\psi_1 \geqq \frac{kt\beta^{1+1/k}}{c}\left[p\left((1-\alpha_1)^{1+1/k} + \frac{\alpha_1^{1+1/k}}{(c/2)^{1/k}}\right) + \bar{p}\frac{1}{((c/2)\alpha_2)^{1/k}}\right] - ky.$$

Hence

$$\psi = \psi_1 - \psi_2 \geqq \frac{kt\beta^{1+1/k}}{c}\left[p\left((1-\alpha_1)^{1+1/k} + \frac{\alpha_1^{1+1/k}}{(c/2)^{1/k}}\right) + \frac{\bar{p}}{((c/2)\alpha_2)^{1/k}}\right]$$
$$- ky - k\left[\frac{y^{1+1/k}}{c(cf/y)^{1/k}} - y\right].$$

Since $t<8f/y$ we conclude, by (2.2), that

$$\psi \geqq \frac{ky^{1+1/k}(1-8/c^{k+1})^{1+1/k}}{ct^{1/k}}\left[p\left((1-\alpha_1)^{1+1/k}+\frac{\alpha_1^{1+1/k}}{(c/2)^{1/k}}\right)+\frac{\bar{p}}{((c/2)\alpha_2)^{1/k}}-\frac{1}{(c/8)^{1/k}(1-8/c^{k+1})^{1+1/k}}\right].$$

As $c=256e$ we can check that $(1-16/c^{k+1})^k \geqq 1-16k/c^{k+1} \geqq \frac{1}{2}$, and hence

$$\left(1-\frac{8}{c^{k+1}}\right)^{1+1/k} \geqq \left(1-\frac{8}{c^{k+1}}\right)^2 \geqq 1-\frac{16}{c^{k+1}} \geqq \left(\frac{1}{2}\right)^{1/k}.$$

Therefore

$$\psi \geqq \frac{ky^{1+1/k}(1-8/c^{k+1})^{1+1/k}}{ct^{1/k}}\left[p\left((1-\alpha_1)^{1+1/k}+\frac{\alpha_1^{1+1/k}}{(c/2)^{1/k}}\right)+\frac{\bar{p}}{((c/2)\alpha_2)^{1/k}}-\frac{1}{(c/16)^{1/k}}\right].$$

To establish (2.3) it suffices to show that

(2.5)
$$h \geqq 0 \quad \text{where}$$
$$h=p\left[(1-\alpha_1)^{1+1/k}+\frac{\alpha_1^{1+1/k}}{(c/2)^{1/k}}\right]+\frac{\bar{p}}{((c/2)\alpha_2)^{1/k}}-\frac{1}{(c/16)^{1/k}}.$$

In Appendix 2 we prove that (2.5) holds, subject to the constraints (2.4) and

(2.6) $$\alpha \leqq \tfrac{1}{2}.$$

In Appendix 3 we establish (2.6). Therefore $\psi \geqq 0$ and (2.3) holds, completing the proof of Subcase 1c and that of Case 1.

Case 2. For all i, $1 \leqq i \leqq s$, $f_i=0$ and $k>1$. Here the next round of comparisons (kth from the end) is performed. Let F be the set of these comparisons. Note that (if $s>1$) F may contain comparisons between members of distinct sets Y_i, and hence the present situation is not necessarily a legal one. Let $f_1, \cdots, f_s$ be the new "f-numbers" of $Y_1, \cdots, Y_s$ (i.e., $f_i=e_i+\frac{1}{2}\bar{e}_i$, where e_i, $\bar{e}_i$ are as in the definition of a legal situation given in § 2.2). Clearly $|F| \geqq \sum_{i=1}^{s} f_i$.

Subcase 2a. $s=1$. In this case the present situation A' is a legal one. For A', the number of remaining rounds is $k'=k-1$, and there is a set $Y=Y_1$ with f-number $f=f_1$, and $|Y|=y$ elements and a set Z. By case (a) of the induction hypothesis $F(k', A') \geqq \varphi_{k-1}(y, f)$. Therefore, to complete the proof for this subcase it suffices to check that

$$f+\varphi_{k-1}(y,f) \geqq \varphi_k(y,0).$$

If $0 \leqq f < y/4$ then

$$f+\varphi_{k-1}(y,f) \geqq \varphi_{k-1}(y,f) \geqq (k-1)\left(\frac{y^{1+1/(k-1)}}{c(c/4)^{1/(k-1)}}-y\right)$$
$$=ky\left[\left(1-\frac{1}{k}\right)\frac{y^{1/(k-1)}}{c(c/4)^{1/(k-1)}}+\frac{1}{k}\cdot 1\right]-ky.$$

By the Arithmetic-Geometric Inequality $\alpha a+\beta b \geqq a^\alpha b^\beta$ for all $\alpha, \beta, a, b \geqq 0$, $\alpha+\beta=1$. Applying it with $\alpha=1-1/k$, $\beta=1/k$ we get

$$f+\varphi_{k-1}(y,f) \geqq ky\left[\frac{y^{(1/(k-1))((k-1)/k)}}{c^{(k-1)/k}c^{1/k}(\frac{1}{4})^{1/k}}\cdot 1^{1/k}\right]-ky$$
$$=\frac{ky^{1+1/k}\cdot 4^{1/k}}{c}-ky \geqq \frac{ky^{1+1/k}}{c}-ky=\varphi_k(y,0),$$

as needed. Otherwise $f > y/4$ and then, by a similar application of the Arithmetic-Geometric Inequality,

$$\begin{aligned} f+\varphi_{k-1}(y,f) &= f+(k-1)\left(\frac{y^{1+1/(k-1)}}{c(cf/y)^{1/(k-1)}}-y\right) \\ &= ky\left[\frac{1}{k}\left(\frac{f}{y}\right)+\left(1-\frac{1}{k}\right)\frac{y^{1/(k-1)}}{c^{1+1/(k-1)}(f/y)^{1/(k-1)}}\right]-(k-1)y \\ &\geqq ky\left(\frac{f}{y}\right)^{1/k}\frac{y^{(1/(k-1))((k-1)/k)}}{c^{(k/(k-1))((k-1)/k)}(f/y)^{1/k}}-(k-1)y \\ &= \frac{ky^{1+1/k}}{c}-(k-1)y \geqq k\left(\frac{y^{1+1/k}}{c}-y\right)=\varphi_k(y,0). \end{aligned}$$

This completes the proof of Subcase 2a.

Subcase 2b. $s>1$. Here the situation is not necessarily legal, as there may be comparisons between distinct sets Y_i. We will show that the average value of the f-number of comparisons of each set Y_i is at least $\varphi_k(y_i, f_i)$. As the total f-number is simply the sum of the f-numbers of the sets Y_i, this will give the desired result. Fix i, $1 \leqq i \leqq s$. Let A_i'' be the legal situation obtained from (the possibly nonlegal one) A' by defining $Z''=N\setminus Y_i$, i.e., by giving the exact rank of each element outside Y_i. (Note that there are many distinct such A_i'', depending on the actual ranks of the elements in $\bigcup_{j\neq i} Y_j$.) As $y_i < \sum_{j=1}^{s} y_j$, we can apply case (b) of the induction hypothesis and conclude that $F(k-1, A_i'') \geqq \varphi_{k-1}(y_i, f_i)$. It follows that any algorithm that sorts in $k-1$ rounds, starting from the situation A' performs on the average at least $\varphi_{k-1}(y_1, f_i)$ (f-number of) comparisons for Y_i, averaging only on those orders in which the ranks outside Y_i are as in the specific choice of A_i''. Averaging over all possible A_i'' and summing over i, $1 \leqq i \leqq s$, we conclude that

$$\begin{aligned} F(k, A) &\geqq \sum_{i=1}^{s} f_i + \sum_{i=1}^{s} \varphi_{k-1}(y_i, f_i) \\ &= \sum_{i=1}^{s} (f_i + \varphi_{k-1}(y_i, f_i)) \geqq \sum_{i=1}^{s} \varphi_k(y_i, 0), \end{aligned}$$

where the last inequality follows from the computation of the previous subcase. This completes the proof of Case 2, and the proof of the induction step.

To complete the proof of Theorem 2.1, it remains to establish the base case of the induction. Clearly this case corresponds to a legal situation A with $k=1$, with a set Z and sets $Y_1, \cdots, Y_s$, where $f_i=0$ for all i (otherwise we are in either Case 1 or Case 2, and can apply the induction hypothesis). By the arguments given in the proof of Subcase 2b, we can reduce the case $s>1$ to the case $s=1$ (otherwise we bound the f-number for each Y_i separately, as in Subcase 2b). Hence we may assume that for A, $k=1$, $s=1$, $Y=Y_1$, $|Y|=y$, $f_1=0$, and Z is a set of elements with known ranks. We must sort Y in one round. Note that since we must complete the sorting in one round we actually have to prove here a worst-case lower bound. If there are two successive ranks in the (known) set of ranks of Y, we must compare each pair of elements of Y. Indeed, suppose that a dispensed comparison is between such two successive elements of Y in the sorted order. The algorithm will clearly fail to determine their relative order. Hence, in this case, $F(1, A) \geqq \binom{y}{2} \geqq \varphi_1(y, 0)$, as needed. Therefore, we may assume that there is at least one element of Z between any two successive elements of Y. Clearly we can assume that there is precisely one element of Z between any two successive elements of Y and that there are no elements of Z that are greater than all members of Y (or smaller than all members of Y), as no additional information can be derived from comparisons to such elements. We can thus assume that $|Z|=|Y|-1=y-1$, the (known) set of ranks of Z is $\{2, 4, \cdots, 2y-2\}$, and the (known) set of ranks of Y is $\{1, 3, \cdots, 2y-1\}$.

Let F be the set of comparisons performed by the algorithm. Suppose $F=F' \cup F''$, where F' are the comparisons between elements of Y and F'' are those between elements of Y and elements of Z. We claim that if a, b are two distinct members of Y and the comparison (= edge) $\{a, b\}$ satisfies $\{a, b\} \notin F'$ then there are at least $y-1=|Z|$ comparisons in F'' that involve a or b. Indeed, if this is false, then there is an element $z \in Z$ such that $\{a, z\}, \{b, z\} \notin F''$. If $\operatorname{rank}(z)=l$, then the algorithm clearly will fail to distinguish between any order in which $\operatorname{rank}(a)=l-1$ and $\operatorname{rank}(b)=l+1$ and the order obtained from it by replacing the ranks of a and b. Thus the claim holds. Summing these comparisons of F'' over all a, b with $\{a, b\} \notin F'$, we obtain at least $(y-1)\cdot(\binom{y}{2}-|F'|)$. In this sum each comparison in F'' is counted at most $y-1$ times, as for every $a \in Y$ there are at most $y-1$ $b \in Y$, $b \neq a$ such that $\{a, b\} \notin F'$. Hence

$$(y-1)|F''| \geqq (y-1)\left(\binom{y}{2}-|F'|\right), \quad \text{i.e., } |F'|+|F''| \geqq \binom{y}{2}.$$

We conclude that the f-number of Y satisfies $f=|F'|+\frac{1}{2}|F''| \geqq \frac{1}{2}\binom{y}{2} \geqq \varphi_k(y, 0)$. This completes the proof of the base case of the induction and establishes Theorem 2.1. □

3. Time lower bounds for randomized parallel sorting algorithms. In this section we derive Theorem 1.1 from Proposition 1.2. The easy fact that for $p \leqq n$ the average time is $\Theta(\log n/(p/n))$ follows from the existence of the sorting network [AKS83a], [AKS83b], together with the $\Omega(n \log n)$ known bound for serial average randomized algorithms. The upper bound in inequality (1.1) implies that for every $p \geqq n$ time $O(\log n/\log(1+p/n))$ is sufficient, even for the worst case of deterministic algorithms. It remains to prove the lower bound $\Omega(\log n/\log(1+p/n))$ for $p \geqq n$. As observed by Yao [Ya77], since any randomized algorithm is simply a probability distribution on deterministic ones, it suffices to establish the same lower bound for the average time of deterministic sorting algorithms with p processors. This does not follow immediately from Proposition 1.2, since in this proposition we considered only algorithms that necessarily stop after k rounds. Hence we need to do some more work. We first need the following simple probabilistic lemma. Let S_n denote the group of all permutations on n elements. For $A \subseteq S_n$ and $g_0 \in S_n$ define $g_0A=\{g_0g \mid g \in A\}$. Also define $q(A)=|A|/|S_n|=|A|/n!$

LEMMA 3.1. *If $A \subseteq S_n$ and $q(A) \leqq \frac{1}{2}$ then for every $s \geqq 1$ there are $g_1, g_2, \cdots, g_s \in S_n$ such that $q(\bigcap_{i=1}^s g_iA) \leqq 1/2^s$.*

Proof. Choose, independently, s (not necessarily distinct) random elements $g_1, g_2, \cdots, g_s$ of S_n. If $h \in S_n$, the probability that $h \in \bigcap_{i=1}^s g_iA$, is precisely $q(A)^s$. Thus, the expected number of elements in $\bigcap_{i=1}^s g_iA$ is $n!q(A)^s \leqq n!/2^s$, and there are $g_1, \cdots, g_s \in S_n$ satisfying the conclusion of Lemma 3.1. □

PROPOSITION 3.2. *Suppose there is a deterministic algorithm M that sorts n elements with p processors in expected time T. Then, for every $s \geqq 1$, there is a deterministic algorithm that sorts n elements in $2T+\lceil T \rceil \leqq 4T$ rounds (and necessarily stops after these $4T$ rounds) with at most $2Tps+1/2^s\lceil T \rceil n^{1+b/T}$ average number of comparisons, where $b>0$ is the constant from (1.1).*

Proof. Let N be the set of elements we have to sort. Let A be the set of all permutations of N that M fails to sort in $\leqq 2T$ rounds. Clearly, $q(A) \leqq \frac{1}{2}$. By Lemma 3.1 there exist $g_1, \cdots, g_s \in S_n$ such that $q(\bigcap_{i=1}^s g_iA) \leqq 1/2^s$. Let M'' be the algorithm in which s copies of M run simultaneously for $\lfloor 2T \rfloor$ rounds, where the ith copy runs on the elements of N permuted according to g_i^{-1}. Clearly M'' finds the order of N, unless this order corresponds to a permutation in $B = \bigcap_{i=1}^{s} g_i A$. But this happens only on a $1/2^s$-fraction of the orders. Let M' be the algorithm that consists of M'', and if M'' fails it uses the best deterministic algorithm for sorting N in $\lceil T \rceil$ rounds. By (1.1) this part takes, in the worst case, at most $\lceil T \rceil n^{1+b/T}$ additional comparisons. This completes the proof. □

Proof of Theorem 1.1. As observed in the beginning of this section, we only have to establish an $\Omega(\log n/\log(1+p/n))$ lower bound for the average time of deterministic algorithms for sorting n elements with $p \geqq n$ processors. By Proposition 1.2 there exists a (small) constant $\frac{1}{2} > c > 0$ such that the average number of comparisons in any deterministic algorithm that sorts $n \geqq 2$ elements in $k \leqq \log n$ rounds is at least

$$ckn^{1+1/k}. \tag{3.1}$$

Let $b \geqq 1$ be the (large) constant from inequality (1.1). Let d be the (small) positive constant defined by

$$\frac{1}{16d} = 1 + \log(16b) + \log(1/2c). \tag{3.2}$$

(Notice that the right-hand side is positive, as $c < \frac{1}{2}$, $b \geqq 1$.)

Let M be a deterministic algorithm that sorts n elements with $p \geqq n$ processors in expected time T. To complete the proof we show that

$$T \geqq d \log n/\log(1+p/n). \tag{3.3}$$

If $T \geqq d \log n$ then (3.3) holds (for every $p \geqq n$). Hence, we may assume that $T < d \log n (< b \cdot \log n)$. Define $s = \lceil b \log n/T \rceil$. Clearly $b \log n/T \leqq s \leqq 2b \log n/T$. By Proposition 3.2, there is a deterministic algorithm that sorts n elements in at most $4T$ rounds with at most

$$2Tps + \frac{1}{2^s}\lceil T \rceil n^{1+b/T} \leqq 4Tp\frac{b \log n}{T} + 2Tn$$

average number of comparisons. Hence, by (3.1)

$$4Tp\frac{b \log n}{T} + 2Tn \geqq c \cdot 4Tn^{1+1/(4T)},$$

i.e.,

$$4\frac{p}{n}\frac{b \log n}{T} \geqq 4cn^{1/(4T)} - 2 \geqq 2cn^{1/(4T)},$$

where the last inequality holds since $\log n/4T > 1/4d > -\log(c)$. By taking logarithms we obtain

$$\log\left(\frac{p}{n}\right) \geqq \frac{\log n}{4T} - \log\left(\frac{\log n}{4T}\right) + \log(2c) - \log(16b).$$

As $\log n/4T > 1/4d > 4$ we have

$$\frac{\log n}{4T} - \log\left(\frac{\log n}{4T}\right) \geqq \frac{\log n}{8T}.$$

Also, by (3.2),

$$\frac{\log n}{16T} > \frac{1}{16d} > \log(16b) - \log(2c).$$

These three inequalities imply

$$\log(p/n) > \frac{\log n}{16T},$$

and hence, for $p \geqq n$ the inequality

$$T > \frac{1}{16}\frac{\log n}{\log(p/n)} > d \log n/\log(1+(p/n))$$

holds (provided our assumption $T < d \log n$ holds). This establishes (3.3) and completes the proof. □

4. Concluding remarks. We have shown that the average running time of any comparison (deterministic or randomized) algorithm for sorting n elements with p processors is $\Theta(\log n/\log(1+p/n))$, for all $p \geqq n$, (and is $\Theta(\log n/(p/n))$ for $p \leqq n$.)

This is the first known nontrivial bound for randomized parallel comparison sorting algorithms. It shows that the average time of the best randomized algorithm is not smaller than the worst-case time of the best deterministic algorithm for all p and n, up to a constant factor.

We note that although Proposition 1.2 is mainly used as a tool for deriving this result, it actually gives additional information. This proposition shows that the average number of comparisons in any deterministic algorithm that sorts n elements in $k \leqq \log n$ rounds is $\Omega(kn^{1+1/k})$. As shown in [AAV86] this result is sharp for any fixed k. Note also that as shown in [AA87], the worst-case number of comparisons for such an algorithm is $\Omega(n^{1+1/k}(\log n)^{1/k})$, i.e., it is bigger for every fixed k. Thus we conclude that the average behaviour is somewhat different from the worst-case one for parallel comparison sorting algorithms. Our main result (Theorem 1.1) shows that this difference is, however, very small and shrinks to a constant factor if we fix the number of processors and estimate the running time. It is more than a constant factor if we fix the time and estimate the number of processors.

Our methods supply some results for randomized selection algorithms as well. In particular, we can show that the average number of comparisons needed in any randomized algorithm for finding the maximum of n elements in two rounds is $\Theta(n^{4/3})$, and for doing so in three rounds is $\Theta(n)$. We omit the details.

Appendix 1. We have to show that for $0 < f \leqq y/4$, $\varphi_k(2f, f) + \varphi_k(y-2f, 0) \geqq \varphi_k(y, f)$. That is,

$$k\left(\frac{(2f)^{1+1/k}}{c(c/2)^{1/k}} - 2f\right) + k\left(\frac{(y-2f)^{1+1/k}}{c} - (y-2f)\right) \geqq k\left(\frac{y^{1+1/k}}{c(c/4)^{1/k}} - y\right),$$

or

$$(y-2f)^{1+1/k} + \frac{(2f)^{1+1/k}}{(c/2)^{1/k}} - \frac{y^{1+1/k}}{(c/4)^{1/k}} \geqq 0.$$

Put $\alpha = 2f/y$ and $h(\alpha) = (1-\alpha)^{1+1/k} + \alpha^{1+1/k}/(c/2)^{1/k} - 1/(c/4)^{1/k}$. We have to show that $h(\alpha) \geqq 0$ for all $0 < \alpha \leqq \frac{1}{2}$. This is done by checking that h is convex, $h'(\frac{1}{2}) \leqq 0$, and $h(\frac{1}{2}) \geqq 0$. Hence for all $0 < \alpha \leqq \frac{1}{2}$ $h'(\alpha) \leqq 0$ and therefore $h(\alpha) \geqq h(\frac{1}{2}) \geqq 0$.

h is convex, as it is a sum of convex functions. $h'(\frac{1}{2}) = -(k+1)/k[(\frac{1}{2})^{1/k} - (1/2)^{1/k}/(c/2)^{1/k}] \leqq 0$, as $c \geqq 2$.

$$h\left(\frac{1}{2}\right) = \left(\frac{1}{2}\right)^{1+1/k}\left[1 + \left(\frac{2}{c}\right)^{1/k} - 2\left(\frac{8}{c}\right)^{1/k}\right]$$

$$= \left(\frac{1}{2}\right)^{1+1/k}\left[\left(1 - \left(\frac{8}{c}\right)^{1/k}\right)^2 + \left(\frac{64}{c^2}\right)^{1/k}\left(\left(\frac{c}{32}\right)^{1/k} - 1\right)\right] \geqq 0,$$

as $c \geqq 32$.

Appendix 2. We have to show that

(2.5)
$$h \geqq 0 \quad \text{where}$$
$$h = p\left[(1-\alpha_1)^{1+1/k} + \frac{\alpha_1^{1+1/k}}{(c/2)^{1/k}}\right] + \frac{\bar{p}}{((c/2)\alpha_2)^{1/k}} - \frac{1}{(c/16)^{1/k}}$$

subject to

(2.4)
$$\alpha_2 \geqq 1, \qquad p\alpha_1 + \bar{p}\alpha_2 = \alpha$$

and

(2.6)
$$\alpha \leqq \tfrac{1}{2}.$$

(Recall that $p, \bar{p}, \alpha, \alpha_1, \alpha_2 \geqq 0$, $p + \bar{p} = 1$, $c = 256e$.) Consider h as a function of α_1, where $\alpha, p, \bar{p}$ are constants and $\alpha_2 = (\alpha - p\alpha_1)/\bar{p}$. Since $\alpha_2 \geqq 1$, $\alpha \leqq \frac{1}{2}$ we have $0 \leqq \alpha_1 \leqq (\alpha - \bar{p})/p = 1 - (1-\alpha)/p$. Clearly $h(\alpha_1)$ is a convex function of α_1, i.e., $h''(\alpha_1) \geqq 0$ for all admissible α_1. Here we prove that $h'(1-(1-\alpha)/p) \leqq 0$ and hence, as $h'(\alpha_1)$ is nonincreasing, we have $h(\alpha_1) \geqq h(1-(1-\alpha)/p)$. We next show that $h(1-(1-\alpha)/p) \geqq 0$ and complete the proof. To check that $h'(1-(1-\alpha)/p) \leqq 0$, observe that

$$h'(\alpha_1) = p\left[-\frac{k+1}{k}(1-\alpha_1)^{1/k} + \frac{k+1}{k}\frac{\alpha_1^{1/k}}{(c/2)^{1/k}}\right] - \bar{p}\frac{c/2(-p/\bar{p})}{k((c/2)\alpha_2)^{1+1/k}}$$

$$= \frac{p}{k}\left[-(k+1)(1-\alpha_1)^{1/k} + (k+1)\frac{\alpha_1^{1/k}}{(c/2)^{1/k}} + \frac{c}{2((c/2)\alpha_2)^{1+1/k}}\right].$$

For $\alpha_1 = 1 - (1-\alpha)/p$ we have $\alpha_2 = 1$; hence (since $0 \leqq p \leqq 1$, $\alpha \leqq \frac{1}{2}$)

$$h'\left(1 - \frac{1-\alpha}{p}\right) = \frac{p}{k}\left[-(k+1)\left(\frac{1-\alpha}{p}\right)^{1/k} + (k+1)\left(\left(1 - \frac{1-\alpha}{p}\right)^{1/k} \Big/ (c/2)^{1/k}\right) + \left(\frac{2}{c}\right)^{1/k}\right]$$

$$\leqq \frac{p}{k}\left[-(k+1)(1-\alpha)^{1/k} + (k+1)\alpha^{1/k}\left(\frac{2}{c}\right)^{1/k} + \left(\frac{2}{c}\right)^{1/k}\right]$$

$$= -\frac{p}{k}\left[(k+1)\left((1-\alpha)^{1/k} - \alpha^{1/k}\left(\frac{2}{c}\right)^{1/k}\right) - \left(\frac{2}{c}\right)^{1/k}\right]$$

$$\leqq -\frac{p}{k}\left[(k+1)\left(\left(\frac{1}{2}\right)^{1/k} - \left(\frac{1}{2}\right)^{1/k}\left(\frac{2}{c}\right)^{1/k}\right) - \left(\frac{2}{c}\right)^{1/k}\right]$$

$$= -\frac{p}{k}\left(\frac{1}{2}\right)^{1/k}\left[(k+1)\left(1 - \left(\frac{2}{c}\right)^{1/k}\right) - \left(\frac{4}{c}\right)^{1/k}\right].$$

The last quantity is clearly negative for $k=1$ (as $c=256e$). For $k \geqq 2$, $(2/c)^{1/k} \leqq (\frac{1}{4})^{1/k} \leqq 1-1/k$, and hence we have

$$h'\left(1-\frac{1-\alpha}{p}\right) \leqq -\frac{p}{k}\left(\frac{1}{2}\right)^{1/k}\left[(k+1)\left(1-\left(1-\frac{1}{k}\right)\right)-1\right] = -\frac{p}{k^2}\left(\frac{1}{2}\right)^{1/k} \leqq 0,$$

as needed. It remains to check that $h(1-(1-\alpha)/p) \geqq 0$. Indeed

$$h\left(1-\frac{1-\alpha}{p}\right) = p\left[\left(\frac{1-\alpha}{p}\right)^{1+1/k} + \left(1-\frac{1-\alpha}{p}\right)^{1+1/k} \Big/ (c/2)^{1/k}\right] + \frac{\bar{p}}{(c/2)^{1/k}} - \left(\frac{16}{c}\right)^{1/k}.$$

Since $(1-x)^\gamma \geqq 1-\gamma x$ for $0 \leqq x \leqq 1$, $\gamma \geqq 1$ this implies

$$\begin{aligned} h\left(1-\frac{1-\alpha}{p}\right) &\geqq \frac{(1-\alpha)^{1+1/k}}{p^{1/k}} + p\frac{1-((1+1/k)(1-\alpha)/p)}{(c/2)^{1/k}} + \frac{1-p}{(c/2)^{1/k}} - \left(\frac{16}{c}\right)^{1/k} \\ &= \frac{(1-\alpha)^{1+1/k}}{p^{1/k}} + \frac{1-(1+1/k)(1-\alpha)}{(c/2)^{1/k}} - \left(\frac{16}{c}\right)^{1/k}. \end{aligned}$$

As $p \leqq 1$ we conclude

$$\begin{aligned} h\left(1-\frac{1-\alpha}{p}\right) &\geqq \left(\frac{2}{c}\right)^{1/k}\left[\left(\frac{c}{2}\right)^{1/k}(1-\alpha)^{1+1/k} + 1 - \left(1+\frac{1}{k}\right)(1-\alpha) - 8^{1/k}\right] \\ &= \left(\frac{2}{c}\right)^{1/k}\left\{(1-\alpha)\left[\left(\frac{c(1-\alpha)}{2}\right)^{1/k} - \left(1+\frac{1}{k}\right)\right] + 1 - 8^{1/k}\right\}. \end{aligned}$$

Since $\alpha \leqq \frac{1}{2}$ and $c=256e$ implies $[(c/2)(1-\alpha)]^{1/k} \geqq (c/4)^{1/k} = (64e)^{1/k} \geqq 64^{1/k} \cdot (1+1/k)$ we have

$$\begin{aligned} h\left(1-\frac{1-\alpha}{p}\right) &\geqq \left(\frac{2}{c}\right)^{1/k}\left\{\frac{1}{2}\left[\left(\frac{c}{4}\right)^{1/k} - \left(1+\frac{1}{k}\right)\right] + 1 - 8^{1/k}\right\} \\ &= \frac{1}{2}\left(\frac{2}{c}\right)^{1/k}\left[\left(\frac{c}{4}\right)^{1/k} + 1 - \frac{1}{k} - 2\cdot 8^{1/k}\right] \\ &= \frac{1}{2}\cdot\left(\frac{2}{c}\right)^{1/k}\left[(1-8^{1/k})^2 + \left(\frac{c}{4}\right)^{1/k} - 64^{1/k} - \frac{1}{k}\right] \\ &\geqq \frac{1}{2}\cdot\left(\frac{2}{c}\right)^{1/k}\left[(1-8^{1/k})^2 + 64^{1/k}\left(1+\frac{1}{k}\right) - 64^{1/k} - \frac{1}{k}\right] \\ &= \frac{1}{2}\cdot\left(\frac{2}{c}\right)^{1/k}\left[(1-8^{1/k})^2 + \frac{1}{k}(64^{1/k}-1)\right] \geqq 0. \end{aligned}$$

This completes the proof of (2.5).

Appendix 3. Here we show $\alpha \leqq \frac{1}{2}$. Recall that $\alpha = a/\beta$, where $a = E(2g) = E(2g'+g'')$, g' is the number of comparisons inside B_1 and g'' is the number of comparisons between B_1 and Z_1. Let F' be the set of all comparisons (= edges) between elements of Y and let F'' be the set of all comparisons between Y and Z whose results do not follow from the known information about the ranks. Put $|F'| = e$, $|F''| = \bar{e}$ and let $f = e + \frac{1}{2}\bar{e}$ be the "f-number" of Y. As the members of B_1 are β random elements of Y we have

$$E(g') = \binom{\beta}{2}\Big/\binom{y}{2}|F'| \leqq \frac{\beta^2}{y^2}e.$$

Consider a fixed edge (= comparison) in F''. Such a comparison compares some member $m_y \in Y$ to some member $m_z \in Z$. The probability that the m_y will be in $\bigcup_{i=1}^{t} B_i$ is, clearly, $\beta t/y$. As the permutation π is chosen randomly, the probability that $m_y \in B_i$, given that $m_z \in Z_i$ (and that $m_y \in \bigcup_{i=1}^{t} B_i$), is $1/t$. Therefore, the expected number of edges from F' that join members of B_i to members of Z_i for some $1 \leqq i \leqq t$ is at most $|F''| \cdot \beta t/y \cdot 1/t = \beta \bar{e}/y$. By the symmetry of the sets B_i this gives $E(g'') \leqq \beta \bar{e}/yt$. Therefore, since $\beta t \leqq y$,

$$a = E(2g) = E(2g' + g'') \leqq 2\frac{\beta^2}{y^2} e + \frac{\beta \bar{e}}{yt} \leqq 2\frac{\beta}{yt}\left(e + \frac{1}{2}\bar{e}\right) = 2\frac{b}{yt} f.$$

Recall that $t \geqq 4f/y$, i.e., $f \leqq ty/4$. This implies $\alpha = a/\beta \leqq 2f/yt \leqq \frac{1}{2}$, as needed.

REFERENCES

[AA88] N. Alon and Y. Azar, *Sorting, approximate sorting, and searching in rounds*, SIAM J. Disc. Meth., 1 (1988), pp. 269–280.

[AAV86] N. Alon, Y. Azar, and U. Vishkin, *Tight complexity bounds for parallel comparison sorting*, in Proc. 27th Annual IEEE Symposium on Foundations of Computer Science, Toronto, Ontario, Canada, 1986, pp. 502–510.

[AHU74] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, London, 1974, pp. 92–93.

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

[AKS83a] M. Ajtai, J. Komlós, and E. Szémerédi, *An $O(n \log n)$ sorting network*, in Proc. 15th Annual ACM Symposium on Theory of Computing, 1983, pp. 1–9.

[AKS83b] ———, *Sorting in $c \log n$ parallel steps*, Combinatorica, 3 (1983), pp. 1–19.

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

[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.

[BHo85] A. Borodin and J. E. Hopcroft, *Routing, merging and sorting on parallel models of computation*, J. Comput. System Sci., 30 (1985), pp. 130–145.

[BT83] B. Bollobás and A. Thomason, *Parallel sorting*, Discrete Applied Math., 6 (1983), pp. 1–11.

[GK87] M. Geréb-Graus and D. Krizanc, *The complexity of parallel comparison merging*, in Proc. 28th Annual IEEE Symposium on Foundations of Computer Science, Los Angeles, CA, 1987, pp. 195–201.

[HH80] R. Häggkvist and P. Hell, *Graphs and parallel comparison algorithms*, Congr. Numer., 29 (1980), pp. 497–509.

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

[HLP59] G. H. Hardy, J. E. Littlewood, and G. Polya, *Inequalities*, Cambridge University Press, Cambridge, 1959.

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

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

[Re81] R. Reischuk, *A fast probabilistic sorting algorithm*, in Proc. 22nd Annual IEEE Symposium on Foundations of Computer Science, Nashville, TN, 1981, pp. 212–219.

[Rei85] J. H. Reif, *An optimal parallel algorithm for integer sorting*, in Proc. 26th Annual IEEE Symposium on Foundations of Computer Science, Portland, OR, 1985, pp. 496–503.

[Th83] C. Thompson, *The* VLSI *complexity of sorting*, IEEE Trans. Comput., 32 (1983), pp. 1171–1184.

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

[Ya77] A. C. C. Yao, *Probabilistic computations: Towards a unified measure of complexity*, in Proc. 18th Annual IEEE Symposium on Foundations of Computer Science, Providence, RI, 1977, pp. 222–227.