SIAM J. COMPUT.
Vol. 16, No. 1, February 1987

© 1987 Society for Industrial and Applied Mathematics
007

# A TIME-SPACE TRADEOFF FOR ELEMENT DISTINCTNESS*

A. BORODIN†[1], F. FICH‡[1], F. MEYER AUF DER HEIDE§[1], E. UPFAL¶ AND A. WIGDERSON*[1]

**Abstract.** In *A time space tradeoff for sorting on non-oblivious machines*, Borodin et al. [J. Comput. System Sci., 22 (1981), pp. 351-364] proved that to sort $n$ elements requires $TS = \Omega(n^2)$ where $T$ = time and $S$ = space on a comparison based branching program. Although element distinctness and sorting are equivalent problems on a computation tree, the stated tradeoff result does not immediately follow for element distinctness or indeed for any decision problem. In this paper, we are able to show that $TS = \Omega(n^{3/2}\sqrt{\log n})$ for deciding element distinctness (or the sign of a permutation).

**Key words.** time-space tradeoffs, computational complexity, time lower bounds, space lower bounds

**AMS(MOS) subject classification.** 68Q25

**1. Introduction.** Time-space tradeoffs are one of the more classical issues in complexity studies. Cobham's [C-66] seminal paper establishes such tradeoffs for the restricted Boolean model of one-tape Turing machines. A number of time-space tradeoffs were established for both the Boolean and arithmetic circuit models (see Tompa [T-80]). Within these models, merging was essentially as difficult as sorting. For the problem of sorting $\{x_1, x_2, \cdots, x_n\}$ from an arbitrary linearly ordered set, the appropriate model is the comparison based branching program. Such programs are labelled directed acyclic graphs derived from comparison computation trees by identifying common subtrees. Following Cobham, space (or capacity) is measured as the $\log_2$ (# *of nodes*) in the program and time (as for computation trees) is the length of the longest path. Borodin, Fischer, Kirkpatrick, Lynch and Tompa [B-81] established the "near optimal" bound of $TS = \Omega(n^2)$ for sorting but were not able to establish a similar result for any decision problem, conjecturing that such a result should hold for the problem of determining element distinctness (i.e. $f(x_1, x_2, \cdots, x_n) = true$ iff $x_i \neq x_j$ for all $i \neq j$).

We are not able to establish the result as conjectured, but we are able to show $TS = \Omega(n^{3/2})$. Since there is no need to output during the computation, we need to find an appropriate notion of progress and show how this progress is constricted by a bound on the space. It turns out that once the appropriate progress notion is made, the basic outline of proof parallels the development in the sorting result.

**2. The model and proof of the main result.** Let $\{x_1, x_2, \cdots, x_n\}$ be elements chosen from an arbitrary linear order $[D, \leqq]$. A comparison branching program is a labelled directed acyclic graph. Each nonsink node has outdegree three and is labelled by a comparison $x_i : x_j$, with one branch for each of the three possible outcomes $<$, $=$, $>$. The sinks are labelled accept and reject. An input $\langle x_1, x_2, \cdots, x_n\rangle \in D^n$ follows a path in a program $P$ in the obvious way determined by the comparisons, and we define acceptance of a set $L \subset D^n$ as usual.

We consider the set $L = \{\langle x_1, x_2, \cdots, x_n\rangle | x_i \neq x_j\}$. We say that $\langle x_1, x_2, \cdots, x_n\rangle \in L$ is ordered by the permutation $\pi$ if $x_{\pi(1)} < x_{\pi(2)} < \cdots < x_{\pi(n)}$. It is clear that $\pi$ determines

* Received by the editors August 29, 1985; accepted for publication (in revised form) March 13, 1986.
† University of Toronto, Toronto, Ontario, Canada M5S1A4.
‡ University of Washington, Seattle, Washington 98195.
§ Johann Wolgang Goethe Universität Frankfurt a. M., Federal Republic of Germany.
¶ IBM Research Laboratory, San Jose, California 95120.
* Mathematical Science Research Institute, Berkeley, California 94720.
[1] This work was done while the authors were visiting the IBM Research Laboratory, San Jose, California 95120.

the comparison path $\tau$ that $\langle x_1, x_2, \cdots, x_n\rangle$ follows in any comparison tree or branching program. Clearly, $\tau$ must terminate in an accepting node. Moreover, it is easy to see that $\tau$ must contain a comparison for every "adjacent" pair $x_{\pi(i)}:x_{\pi(i+1)}$, $1 \leqq i \leqq n-1$. Otherwise, we can set $x_{\pi(i+1)} = x_{\pi(i)}$ and still follow the same path $\tau$ (and erroneously accept). The number of adjacent pairs tested on a path will measure the progress for any $\pi$. Once we establish the following main lemma, we can follow the same proof structure as in Borodin et al. [B-81].

MAIN LEMMA. *Let $T$ be a comparison computation tree of height (= time) $t$. If $t \leqq \sqrt{ns/16e}$ then for all $S$ the fraction $p$ of input permutations $\pi$ for which $\pi$ follows a path containing more than $S$ comparisons of adjacent pairs is bounded by $p \leqq (\frac{1}{4})^S$.*

*Proof of Main Lemma.* Let $\tau$ be a computation path of length $t \leqq \sqrt{sn/16e}$. There are at most $r \leqq 2t$ elements $x_i$ that are involved in some comparison in $\tau$. $\tau$ determines a partial order on $r$ elements. We will compute the fraction of permutations $\pi$ following $\tau$ for which we have made at least $S$ comparisons $x_{\pi(i)}:x_{\pi(i+1)}$. Let $\sigma$ be any total order of the $r$ elements consistent with $\tau$. We bound the fraction for each $\sigma$ as follows:

There are at most $\binom{n}{r}$ ways to assign ranks to the accessed elements. We now bound the number of rank assignments under the constraint that at least $S$ adjacent pairs have been tested. There are $\binom{t}{S}$ ways to choose the $S$ pairs and then $\binom{n}{(r-S)}$ ways to assign ranks to the $r$ elements. (If $x_j < x_k$ is one of the $S$ adjacent pairs, fixing the rank of $x_j$ fixes the rank of $x_k$.)

Hence,

$$p \leqq \frac{\binom{t}{S}\binom{n}{(r-S)}}{\binom{n}{r}} = \binom{t}{S}\frac{r(r-1)\cdots(r-S+1)}{(n-r+S)\cdots(n-r+1)}$$

$$\leqq t^S \frac{r^S}{(n-r)^S} \leqq (4e)^S\left(\frac{t^2}{sn}\right)^S \leqq \left(\frac{1}{4}\right)^S. \qquad \square$$

With the main lemma now established, we proceed exactly as in Borodin et al. [B-81].

THEOREM. *Let $P$ be a time $T$, space $S$ comparison branching program for deciding element distinctness on $n$ elements. Then $T^2S = \Omega(n^3)$ and $TS = \Omega(n^{3/2}\sqrt{\log n})$.*

*Proof.* Consider $P$ in stages, where each stage represents $t = \frac{1}{4}\sqrt{n}$ steps. Without loss of generality, $S \geqq \log n$, since each of the possible comparisons appears at least once in the program, which therefore has at least $\binom{n}{2}$ nodes.

Let $q_i$ be the fraction of input permutations (of distinct elements) for which $P$ has compared at least $iS$ adjacent pairs by the end of the $i$th stage. Using the main lemma, we will show that $q_i \leqq i(\frac{1}{2})^S$. Hence $q_i < 1$ for $i \leqq n/S$. This in turn implies that there must be at least $n/S$ stages so that $T \geqq (n/4S)\sqrt{n}$. Otherwise, there will be a permutation for which some adjacent pair has not been tested, forcing a contradiction as previously explained. Thus $T^2S = \Omega(n^3)$. As $S \geqq \log n$, $TS = \Omega(n^{3/2}\sqrt{\log n})$.

To establish the claimed bound for $q_i$, we can consider each of the $2^S$ nodes at the end of stage $i$ to be the root of a *subtree* of height $t$. That is, expand the branching program for stage $i+1$ into at most $2^S$ computation trees. For each such tree at most a fraction $(\frac{1}{4})^S$ of all $n!$ permutations can have more than $S$ adjacent pairs tested. (Note that this is independent of the permutations that actually arrived at the root of this tree.) By the main lemma, the fraction of permutation for which $P$ compared at least $S$ adjacent pairs in the $j$th step is $\geqq (\frac{1}{2})^S$. Hence, the fraction of permutations for which $P$ compared at least $iS$ adjacent pairs in the first $i$ stages is $i(\frac{1}{2})^S$. $\square$



It is easy to see that the sa
the sign of a permutation. In th
can change the ordering of som

**3. The obvious open questio**
upper bound of $TS = O(n^2 \log$
even for sorting [B-81]), the m
see why the existing proof techni
the space bound at the start of
the sorting lower bound), then
only constant width at the begin
$O(\log n)$ blocks of size $O(n/\log$
distinctness by sorting.) We still
ness but any proof will have to

Perhaps the most importan
program model. In Borodin an
to the setting of sorting $n$ intege
of $2n \log n$ bits. This lower bou
of progress, namely how many
bound exploits the fact that w
that of "adjacent pairs tested."
Cook [BC-82]) this concept doe
one notes that in the compariso
sort from the comparison path
Reingold [R-72]) whereas in t
(simply look at each input onc
time-space lower bound for an

**Acknowledgments.** We wish
on an earlier version of this pap
lower bound.

[BC-82] A. BORODIN AND S. COO
*computation*, this Journal,
[B-81] A. BORODIN, M. FISCHER,
*for sorting on non-obliviou*
[C-66] A. COBHAM, *The recognitio*
Watson Research Center,
[R-72] E. REINGOLD, *On the optir*
649-659.
[T-80] M. TOMPA, *Time-space trade*
J. Comp. System Sci., 20

any comparison tree or branching
node. Moreover, it is easy to see
t" pair $x_{\pi(i)}:x_{\pi(i+1)}$, $1 \leqq i \leqq n-1$.
he same path $\tau$ (and erroneously
th will measure the progress for
, we can follow the same proof

*tion tree of height (= time) t. If*
*utations $\pi$ for which $\pi$ follows a*
*airs is bounded by $p \leqq (\frac{1}{4})^S$.*
ath of length $t \leqq \sqrt{sn/16e}$. There
e comparison in $\tau$. $\tau$ determines
tion of permutations $\pi$ following
$x_{\pi(i)}:x_{\pi(i+1)}$. Let $\sigma$ be any total
e fraction for each $\sigma$ as follows:
he accessed elements. We now
nstraint that at least $S$ adjacent
he $S$ pairs and then $\binom{n}{r-S}$ ways
the $S$ adjacent pairs, fixing the

$$\frac{-S+1)}{(n-r+1)}$$

$$\left(\frac{t^2}{sn}\right)^S \leqq \left(\frac{1}{4}\right)^S. \quad \square$$

ed exactly as in Borodin et al.

*branching program for deciding*
*nd $TS = \Omega(n^{3/2}\sqrt{\log n})$.*
presents $t = \frac{1}{4}\sqrt{n}$ steps. Without
e comparisons appears at least
odes.
distinct elements) for which $P$
the $i$th stage. Using the main
$\leqq n/S$. This in turn implies that
$\sqrt{n}$. Otherwise, there will be a
tested, forcing a contradiction
$n$, $TS = \Omega(n^{3/2}\sqrt{\log n})$.
nsider each of the $2^S$ nodes at
That is, expand the branching
es. For each such tree at most
than $S$ adjacent pairs tested.
actually arrived at the root of
tion for which $P$ compared at
e fraction of permutations for
$i$ stages is $i(\frac{1}{2})^S$. $\square$



It is easy to see that the same proof establishes the same bound for determining the sign of a permutation. In this case, if not all adjacent pairs have been tested, we can change the ordering of some pair and thereby change the sign.

**3. The obvious open questions.** Since we do not know how to improve the known upper bound of $TS = O(n^2 \log n)$ for element distinctness (this upper bound holding even for sorting [B-81]), the most obvious open question is whether $TS = \Omega(n^2)$. To see why the existing proof techniques do not extend, we note that we are only exploiting the space bound at the start of each stage. If we allow a stage to be $O(n)$ steps (as in the sorting lower bound), then we can decide distinctness in $O(\log^2 n)$ stages using only constant width at the beginning of each stage. (Simply partition the elements into $O(\log n)$ blocks of size $O(n/\log n)$, and then each stage can test a pair of blocks for distinctness by sorting.) We still conjecture that $TS = \Omega(n^2)$ holds for deciding distinctness but any proof will have to frequently exploit the space bound.

Perhaps the most important extension of our results is to the Boolean branching program model. In Borodin and Cook [BC-82] the sorting lower bound is extended to the setting of sorting $n$ integers in the range $[1, n^2]$, the input represented by a string of $2n \log n$ bits. This lower bound is made possible since we have an explicit notion of progress, namely how many ranks have been established. Our comparison lower bound exploits the fact that we could also find a "monotone" concept of progress, that of "adjacent pairs tested." For the Boolean or "R-way model" (see Borodin and Cook [BC-82]) this concept does not apply. The difficulty of this issue is realized when one notes that in the comparison model, "silent sorting" (i.e. being able to infer the sort from the comparison paths) is equivalent to deciding element distinctness (see Reingold [R-72]) whereas in the Boolean or R-way model, silent sorting is trivial (simply look at each input once). Thus far, we still have not established a nontrivial time-space lower bound for any specific decision problem in the Boolean setting.

**Acknowledgments.** We wish to thank D. Coppersmith and M. Tompa for comments on an earlier version of this paper including an improvement of $\sqrt{\log n}$ to our previous lower bound.

## REFERENCES

[BC-82] A. Borodin and S. Cook, *A time-space tradeoff for sorting on a general sequential model of computation*, this Journal, 11 (1982), pp. 287–297.

[B-81] A. Borodin, M. Fischer, D. Kirkpatrick, N. Lynch and M. Tompa, *A time-space tradeoff for sorting on non-oblivious machines*, J. Comput. System Sci., 22 (1981), pp. 351–364.

[C-66] A. Cobham, *The recognition problem for the set of perfect squares*, Research Paper RC-1704, IBM Watson Research Center, Yorktown Heights, NY, April 1966.

[R-72] E. Reingold, *On the optimality of some set algorithms*, Assoc. Comput. Mach., 19 (1972), pp. 649–659.

[T-80] M. Tompa, *Time-space tradeoffs for computing functions using connectivity properties of their circuits*, J. Comp. System Sci., 20 (1980), pp. 118–132.