ON THE AVERAGE-CASE COMPLEXITY OF SELECTING THE k-th BEST

by

Andres C. Yao and F. Frances Yao

STAN-CS-79-737
April 1979

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

On The Average-case Complexity of Selecting the k-th Best[*]

Andrew C. Yao[+] and F. Frances Yao

Computer Science Department
Stanford University
Stanford, California 94305

Abstract.

Let $\bar{V}_k(n)$ be the minimum average number of pairwise comparisons needed to find the k-th largest of n numbers $(k \geq 2)$, assuming that all n! orderings are equally likely. D. W. Matula proved that, for some absolute constant c, $\bar{V}_k(n)-n \leq c\ k \log \log n$ as $n \to \infty$. In the -present-paper, we show that there exists an absolute constant $c' > 0$ such that $\bar{V}_k(n)-n \geq c'\ k \log \log n$ as $n \to \infty$, proving a conjecture of Matula.

Keywords: algorithm, average-case, binary tree, comparison, complexity, decision tree, selection.

[*] This research was supported in part by National Science Foundation grants MCS-72-03752 A03 and MCS-77-05313.

[+] Part of this work was done while this author was visiting Bell Laboratories, Murray Hill, New Jersey 07974.

1. Introduction.

The problem of selecting the k-th largest in a set of n numbers by pairwise comparisons has been a subject of considerable interest (e.g. Knuth [6][8]). Two particularly interesting situations are the fixed-k case ($n \to \infty$) and the median-finding problem ($k = \lceil n/2 \rceil$). Let $V_k(n)$ denote the complexity of selection in the worst case, and $\bar{V}_k(n)$ the average-case complexity assuming that all n! permutations are equally likely. Table 1 summarizes the known results.*/

	fixed k ($n \to \infty$)	median+/
$V_k(n)$	$V_k(n)-n = (k-1) \lg n + f(k)$ [2] [4] [5] [10]	$3n \geq V_{n/2}(n) \geq 1.75n$≠/ [10] [11]
$\bar{V}_k(n)$	$ck \ln \ln n > \bar{V}_k(n)-n > ?$ [9]	$1.5n \geq \bar{V}_{n/2}(n) \geq 1.375n$ [1]

Table 1. A summary of known results on selection problems.

As seen from the table, no good lower bound is known for the fixed-k behavior of $\bar{V}_k(n)$. It is not even known whether $\bar{V}_2(n)-n \to \infty$ as $n \to \infty$ [6][8]. Sobel conjectured [8] $\bar{V}_2(n)-n$ to be of the order $\log n$, as is true in the worst-case complexity, But in 1973, Matula [9] devised an elegant algorithm which finds the k-th largest using $n + ck(\ln \ln n)$

*/ In this paper, we use lg to stand for logarithm with base 2.

+/ These results have generalizations for the case $k = \alpha n$ with any fixed $0 < \alpha < 1$.

≠/ An improved lower bound of $(11/6)n$ was claimed in [12].

comparisons on the average; and he conjectured that the $k(\ln \ln n)$ term cannot be further reduced. In this paper, we prove that $\bar{V}_k(n)-n \geq c'k(\ln \ln n)$, thus confirming the conjecture. As a result, $\bar{V}_k(n)-n$ is determined to within a constant factor asymptotically.

Main Theorem. For every integer $k \geq 2$, there exists a number N_k such that $\bar{V}_k(n)-n \geq \frac{1}{2} k(\ln \ln n - \ln k - 9)$ for all $n \geq N_k$.

In Section 2 some basic concepts are introduced, In Section 3 we illustrate certain aspects of the proof by showing a weaker form of the theorem in the case $k = 2$, under a severe "regularity" constraint on the class of allowed algorithms. In Section 4 we examine the difficulties encountered in extending the discussion to include non-regular algorithms, We then introduce some new concepts and prove a crucial result (the Limited-Anomaly Theorem) to prepare for the proof of the Main Theorem, which is completed in Section 5.

2. The Accounting Schemes.

An algorithm for selecting the k-th largest of n (distinct) elements $X = \{x_1, x_2, \ldots, x_n\}$ is a binary decision tree T [8]. Associated with each internal node v is a comparison between two elements x_i, x_j. We will say " v compares x_i, x_j ", and use the notation $comp(v) = (x_i : x_j)$. The branching at v is determined by whether $x_i < x_j$ or $x_i > x_j$. By analogy with a tennis tournament that selects the k-th best of n players, we will freely use in this paper descriptions such as " x_i defeats x_j " (if $x_i > x_j$), " x_i is undefeated (so far) ", etc.

Any particular ordering σ satisfied by the input, i.e., $x_{\sigma(1)} > x_{\sigma(2)} > \cdots > x_{\sigma(n)}$, determines a path from the root to a leaf in T. Let $S(\sigma)$ denote the sequence of internal nodes on this path; and let $s(\sigma) = |S(\sigma)|$, the number of comparisons made. The average cost of T is

$$\mathrm{COST}(T) = \frac{1}{n!} \sum_{\sigma} s(\sigma) . \tag{2.1}$$

The average-case complexity $\bar{V}_k(n)$ of selecting the k-th best of n is the minimum cost COST(T) among all decision trees. Without loss of generality, we consider only algorithms that make no redundant comparisons (i.e., comparisons whose results can be deduced from comparisons made previously).

Let T be any algorithm. We consider two types of non-crucial comparisons: for each input ordering σ, let $S_1(\sigma)$ be the set of comparisons made by T in which the loser has been defeated previously, and $S_2(\sigma)$ the set of comparisons involving at least one player ranking in the top k-l . We shall write $s_i(\sigma) = |S_i(\sigma)|$ $(i = 1,2)$. Note that a comparison can be in both $S_1(\sigma)$ and $S_2(\sigma)$. As each player except the top k must encounter a first defeat, we have

$$s(\sigma) \geq n-k+s_1(\sigma) \quad . \tag{2.2}$$

Also, because each player not in the top k must lose to some player ranking below the top (k-l) , we have

$$s(\sigma) \geq n-k+s_2(\sigma) \quad . \tag{2.3}$$

Formulas (2.1), (2.2), (2.3) lead to

$$\mathrm{COST}(T) \geq n-k+\frac{1}{n!}\sum_{\sigma} s_1(\sigma) \quad , \tag{2.4}$$

and

$$\mathrm{COST}(T) \geq n-k+\frac{1}{2}\frac{1}{n!}\sum_{\sigma} (s_1(\sigma)+s_2(\sigma)) \quad . \tag{2.5}$$

We will transform (2.5) into another form. For each internal node v , let $q_i(v)$ $(i = 1,2)$ be the probability that comp(v) is in $S_i(\sigma)$. Precisely, if we let $r(v) = \{\sigma \mid s(\sigma) \text{ contains } v\}$ and $\Gamma_i(v) = \{\sigma \mid \sigma \in \Gamma(v), \text{comp}(v) \in S_i(\sigma)\}$ $(i = 1,2)$, then

$$q_i(v) = \frac{|\Gamma_i(v)|}{|\Gamma(v)|} \quad .$$

We define further

$$q(v) = q_1(v) + q_2(v) \quad ,$$

and

$$\alpha(\sigma) = \sum_{v \in s(a)} q(v) \quad .$$

Then

$$\sum_{\sigma} (s_1(\sigma) + s_2(\sigma)) = \sum_{v \in T} (|\Gamma_1(v)| + |\Gamma_2(v)|)$$

$$= \sum_{v \in T} |\Gamma(v)| q(v)$$

$$= \sum_{\sigma} \sum_{v \in S(\sigma)} q(v)$$

$$= \sum_{\sigma} \alpha(\sigma) \quad . \tag{2.6}$$

We obtain from (2.5) and (2.6),

$$\text{COST}(T) \geq n - k + \frac{1}{2} \frac{1}{n!} \sum_{\sigma} \alpha(\sigma) \quad . \tag{2.7}$$

we collect (2.4) and (2.7) in the following lemma.

Lemma 2.1.

$$\text{COST}(T) \geq n - k + \frac{1}{n!} \sum_{\sigma} s_1(\sigma) \quad , \tag{2.8}$$

$$\text{COST}(T) \geq n - k + \frac{1}{2} \frac{1}{n!} \sum_{\sigma} \alpha(\sigma) \quad . \tag{2.9}$$

We can think of the two formulas in the above lemma as two counting methods for the comparisons. The first one is direct counting, while the other is distributive counting as the cost is "distributed" to the internal nodes of the decision tree. To illustrate the utility of these alternative counting methods, we can combine the two formulas to obtain

$$\text{COST}(T) \geq n - k + \frac{1}{4} \frac{1}{n!} \sum_{\sigma} (s_1(\sigma) + \alpha(\sigma)) \quad . \tag{2.10}$$

Our aim will be, roughly speaking, to show that for any permutation σ ,

$$s_1(\sigma) + \alpha(\sigma) \geq \text{const.} \times k \ln \ln n \qquad (2.11)$$

That is, for any computation sequence $S(\sigma)$, either itself contains a large number $s_1(\sigma)$ of non-crucial comparisons, or it will effect a large number $\alpha(\sigma) = \sum_{v \in S(\sigma)} q(v)$ of non-crucial comparisons distributed over other paths. However, in the proof we shall not be using (2.10) and (2.11), but rather Lemma 2.1 itself, in order to obtain better coefficients of $k \ln \ln n$ in the lower bounds.

Remark. The quantities $s(\sigma), s_i(\sigma), \alpha(\sigma), \ldots$ all depend on T ; we have suppressed this dependence in our notations for simplicity.

3. Regular Algorithms.

3.1 Introduction.

In this section we shall prove a weaker form of the Main Theorem for $k = 2$, under certain "regularity" constraints on the algorithms under consideration.

We begin with a discussion about general algorithms. Let T be any decision tree algorithm selecting the k-th largest of $X = \{x_1, x_2, \ldots, x_n\}$. One can view the computation process for any input ordering σ as building up successively larger partial orders on X, Formally we associate with each node v in T a partial order P(v), which is the transitive closure of all the relations $x_i > x_j$ obtained on the path from the root of T to v (prior to performing the comparison at v). We call $comp(v) = (x_i : x_j)$ a joining comparison if x_i and x_j belong to different connected components in P(v). At each leaf ℓ, $P(\ell)$ must contain only a single component, otherwise the relative order of elements in different components can change the identity of the k-th largest element. Thus, there are exactly n-l joining comparisons $comp(v)$ in the sequence $v \in S(\sigma)$ for any σ; we denote the subsequence of these nodes v by $S'(\sigma)$.

Clearly x is a maximal element in the partial order P(v) if and only if x is yet undefeated. A component C of a partial order is said to be anomalous if C has more than one maximal element. A maximal element x in P(v) is anomalous if x is in an anomalous component, and normal otherwise. A partial order is anomalous if it contains an anomalous component. Figure 1 shows an anomalous partial order with C_1 being an anomalous component, x_2 a normal element, and x_1, x_3 two anomalous elements.

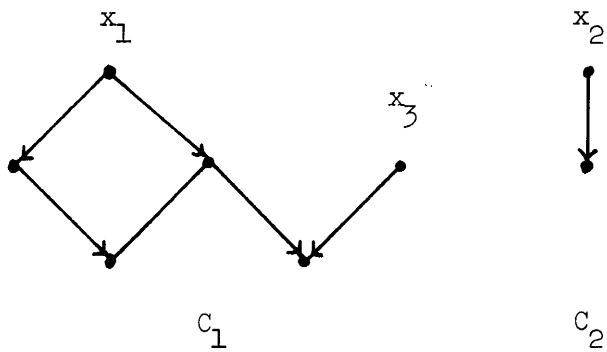

Figure 1. An anomalous partial order

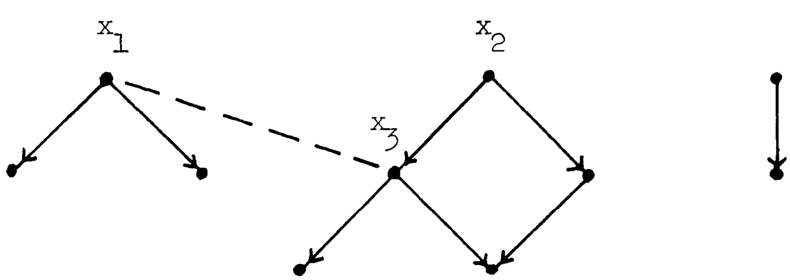

Figure 2. Creation and removal of an anomaly.

We now define the notion of regular algorithms, in which the choice of a comparison comp(v) is restricted by the current partial order P(v).

<u>Definition 3.1</u>. An algorithm T is <u>regular</u> if no joining comparison can involve an anomalous (maximal) element.

In particular, any algorithm that removes anomalous partial order as soon as they occur is regular. For instance, suppose the current partial order P(v) is as shown in Figure 2 and comp(v) = $(x_1 : x_3)$ is performed with result $x_1 \; x_3$, thereby creating an anomalous partial order. By choosing the next comparison to be $(x_1 : x_2)$, we can immediately remove the anomaly independent of the outcome. Matula's algorithm [9] for $k = 2$ is of this type.

The rest of this section is devoted to proving the following result.

<u>Theorem 3.1</u>. Let T be a regular algorithm for selecting the second largest element of $\{x_1, x_2, \ldots, x_n\}$. Then

$$\mathrm{COST}(T) - n \geq \frac{1}{2} \ln \ln n - 6 ,$$

3.2 <u>Some Properties of Binary Trees</u>.

We digress to discuss some useful facts about binary trees.

Let M be a binary tree. We use M_I to denote the set of internal nodes. For each node u, we use notations father[u], brother[u], ℓson[u], rson[u] for the father, brother, leftson, rightson of u, respectively. Let D(u) be the set of internal-node-descendants of u, and $D_L(u)$ the set of leaf-descendants (u is also considered to be a descendant of u). The weight $w(u)$ is the number of leaf-descendants of u ; thus $w(u) = |D_L(u)| = |D(u)|+1$, and for any leaf u , $w(u) = 1$. The external path length is defined as $E(M) = \sum_{u \in M_I} w(u)$.

Lemma 3.2. Let M be any binary tree with n leaves, then $E(M) \geq n(\lg n - 1)$.

Proof. From Knuth [7, Section 2.3.4.5 eqs. (3) and (4)], one has $E(M) \geq n\lfloor \lg n \rfloor - 2n + 2 + 2(n-1) \geq n(\lg n - 1)$. □

Let $H_n = \sum_{1 \leq i \leq n} 1/i$ be the Harmonic numbers (see [7]). It is clear that

$$H_n - H_{n'} = \frac{1}{n'+1} + \frac{1}{n'+2} + \cdots + \frac{1}{n} \geq \int_{n'+1}^{n+1} \frac{1}{x}\, dx ,$$

therefore

$$H_n - H_{n'} > \ln\left(\frac{n+1}{n'+1}\right) \qquad \text{for } n \geq n' \geq 0 . \tag{3.1}$$

Definition 3.2. Let M be a binary tree. A subset of nodes V is called a cross section of M if root $\notin V$ and the following condition is true: For any two distinct $u_i, u_j \in V$, $\text{father}[u_i] \neq \text{father}[u_j]$ and u_i , u_j have no common descendants.

Lemma 3.3. If V is a cross section of a binary tree M with n leaves, then

$$\sum_{u \in V} \frac{w(u)}{w(\text{brother}[u])} \geq \ln\left(\frac{n+1}{n-W+1}\right) ,$$

where $W = \sum_{u \in V} w(u)$.

Proof. For each node u of M, use u' to denote brother[u] when it exists (i.e., when $u \neq$ root). Let depth(u) be the distance from the root to a node u, with depth(root) = 0, We sort the nodes in V in decreasing order of the depth as $u_1, u_2, \ldots, u_t$; i.e., $i < j$ implies $\mathrm{depth}(u_i) \geq \mathrm{depth}(u_j)$.

Fact A. For any $i < j$, u_i' and u_j have no common descendants.

Proof of Fact A. The case $i = j$ is trivial, as u_i' and u_j are brothers. Assume $i < j$, which implies $\mathrm{depth}(u_i') = \mathrm{depth}(u) \geq \mathrm{depth}(u_j)$. If u_i' and u_j have any common descendants, then u_i', and hence u_1, must be a descendant of u_j . But this is ruled out since V is a cross section. ▢

From Fact A, we have for $1 \leq i \leq t$,

$$w(u_i') < n - \sum_{i \leq j \leq t} w(u_j) = n - W + \sum_{1 < j < i} w(u_j) .$$

Let $W(i) = \sum_{1 \leq j \leq i} w(j)$, then

$$\frac{w(u_i)}{w(u_i')} \geq \frac{w(u_i)}{n - W + W(i-1)}$$

$$\geq \sum_{1 \leq j \leq w(u_i)} \frac{1}{n-W+ W(i-1) + j} , \quad 1 \leq i \leq t .$$

Therefore

$$\sum_{u \in V} \frac{w(u)}{w(u')} = \sum_{1 \le i \le t} \frac{w(u_i)}{w(u_i')}$$

$$> \sum_{1 \le j \le W} \frac{1}{n-W+j}$$

$$= H_n - H_{n-W} \; .$$

Lemma 3.3 then follows from formula (3.1).

3.3 Merge-trees and the Proof of Theorem 3.1.

Let T be a regular algorithm that selects the second best of n players. We-shall show that, for any σ ,

$$\sum_{v \in S'(\sigma)} q(v) \ge \ln \ln n - 7 \qquad (3.2)$$

This immediately implies Theorem 3.1, since by Lemma 2.1,

$$\mathrm{COST}(T) \ge n - 2 + \frac{1}{2} (\ln \ln n - 7)$$

$$\ge n + \frac{1}{2} \ln \ln n - 6$$

We first state a useful fact.

<u>Fact B</u>. Let $a_1, a_2, \ldots, a_t$ be positive numbers. Then

$$\sum_{1 \le i \le t} a_i \lg a_i \ge t(\bar{a} \lg \bar{a}) \; ,$$

when $\bar{a} = \left(\sum_i a_i \right) / t$.

<u>Proof</u>. The function $x \lg x$ isconvexfor $x > 0$. $\square$

The basis for proving (3.2) is the following bound on q(v) .

Lemma 3.4. Let $v \in T$ and $\mathrm{comp}(v) = (x_i : x_j)$ a joining comparison between elements in two components of sizes c_1, c_2, respectively. Then

$$q(v) \geq \min\left\{\frac{c_1}{c_1+c_2}, \frac{c_2}{c_1+c_2}, \frac{c_1+c_2}{n}\right\} .$$

Proof. Recall that $q(v) = q_1(v) + q_2(v)$. There are four cases. If x_i and x_j are both undefeated, then $q_2(v) \geq (c_1+c_2)/n$ as the larger of x_i, x_j will be the largest of all elements with probability $(c_1+c_2)/n$. If neither is undefeated, then $q_1(v) = 1 > c_1/(c_1+c_2)$. If x_i is undefeated and x_j is not, then $q_1(v) =$ (Probability that $x_i > x_j) \geq c_1/(c_1+c_2)$. If x_j is undefeated and x_i is not, then $q_1(v) \geq c_2/(c_1+c_2)$ by the same token. Thus the lemma is true in all cases. ☐

We shall now apply the lower bound on q(v) to prove (3.2). We construct an auxiliary binary tree that represents the successive joining operations performed in $S'(\sigma)$, and then use results obtained in Section 3.2.

Merge-tree. Let σ be an input ordering to algorithm T . We can construct a binary tree $M(\sigma)$ corresponding to $S'(\sigma)$ with the following properties.

(1) $M(\sigma)$ has n leaves labeled by the n input elements $X = \{x_1, x_2, \ldots, x_n\}$.

(2) Each internal node u of $M(\sigma)$ corresponds to a $v \in S'(\sigma)$; the x_i's that are descendants of $\ell son[u]$ and $rson[u]$ respectively form the two components that are joined by the comparison at v .

An example of a merge-tree is shown in Figure 3.

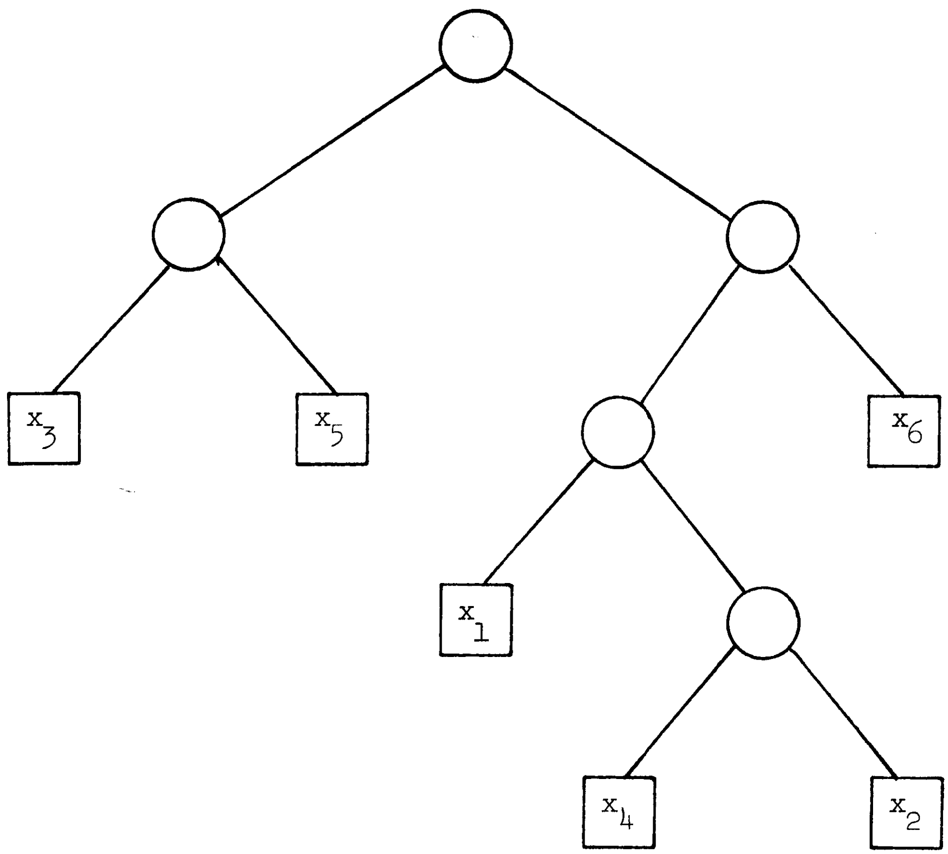

Figure 3. The merge-tree $M(\sigma)$ corresponding to the sequence of joining comparisons $((x_3 : x_5)$, $(x_4 : x_2)$, $(x_1 : x_4)$, $(x_6 : x_2)$, $(x_3 : x_1))$.

Let $C(u)$ denote the subset of X which label the leaf-descendants of u in $M(c)$. Define a function φ on $M(\sigma)_I$, the set of internal nodes of $M(c)$, by letting $\varphi(u) = q(v)$ if u corresponds to $v \in S'(\sigma)$. We wish to prove the following equivalent formula of (3.2).

$$\sum_{u \in M(\sigma)_I} \varphi(u) \geq \ln \ln n - 7 \quad . \tag{3.3}$$

By Lemma 3.4, we have for each $u \in M(\sigma)_I$,

$$\varphi(u) \geq \min\left\{ \frac{w_1}{w_1+w_2}, \frac{w_2}{w_1+w_2}, \frac{w_1+w_2}{n} \right\} , \tag{3.4}$$

where $w_1 = w(\ell son[u])$ and $w_2 = w(rson[u])$. Therefore, Theorem 3.1 will follow from the following result.

<u>Lemma 3.5.</u> Let M be any binary tree with n leaves. For each $u \in M_I$, let $g(u) = \min\{(w_1+w_2)/n, w_1/(w_1+w_2), w_2/(w_1+w_2)\}$ where $w_1 = w(\ell son[u])$ and $w_2 = w(rson[u])$. Then

$$\sum_{u \in M_I} g(u) \geq \ln \ln n - 7 \; .$$

<u>Proof.</u> The proof makes use of the lemmas in Section 3.2. It is given in Appendix A because of its length. $\square$

3.4 Remarks.

The lower bound given in Theorem 3.1 is only about half as large as the corresponding bound in the Main Theorem. This is due to the use of a relatively loose bound for $q(v)$ in Lemma 3.4. A stronger bound for $q(v)$ will be used in the general proof in Section 5, where the regularity constraint is also dropped.

We also wish to point out that (2.8), the first formula in Lemma 2.1, was not used in the above proof, but will be needed later in the proof for the general case.

4. The Limited-Anomaly Theorem.

The arguments in the previous section fail when algorithms are not required to be regular. The important assertion in Lemma 3.4 is no longer true. Consider the partial order $P(v)$ exhibited in Figure 4, and suppose that the next comparison v is between x_1 and an anomalous maximal element x_2. Although the components C_1 and C_2 have sizes 5 and 102 respectively, it is intuitively clear that the probability $q_2(v)$ is less than $(5+102)/n$, as $\max\{x_1,x_2\}$ is unlikely to be the largest among elements in $C_1 \cup C_2$. It will be seen later (Section 5.3) that, in estimating $q_2(v)$, one should use $f(x_2)$, the number of elements in P(v) that are less than (or equal to) x_2 but not less than any other maximal elements, in place of the component size $|C_2|$. In this example $f(x_2) = 4$ and thus $q_2(v) \geq (5+4)/n$, a much weaker lower bound than $(5+102)/n$. Therefore, two complications arise when non-regular algorithms are considered, Firstly, it was previously possible to attach a lower bound to q(v) which depended only on the shape of the associated merge tree; now more details of the partial order $P(v)$ must be taken into account. Secondly, when comparisons involving anomalous elements x_i occur, we may obtain very weak bounds on $q(v)$, if $f(x_i)$ is small. We shall presently prove a result to overcome the second difficulty, by stating that comparisons involving an anomalous maximal element x_i with a small $f(x_i)$ cannot happen too often unless $COST(T)$ is large anyway.

Let P be a partial order on $X = \{x_1,x_2,\ldots,x_n\}$. For each x_i, let $H(x_i)$ be the component containing x_i, and $h(x_i) = |H(x_i)|$. For any maximal element x_i, the fiefdom of x_i, $F(x_i)$ is the set

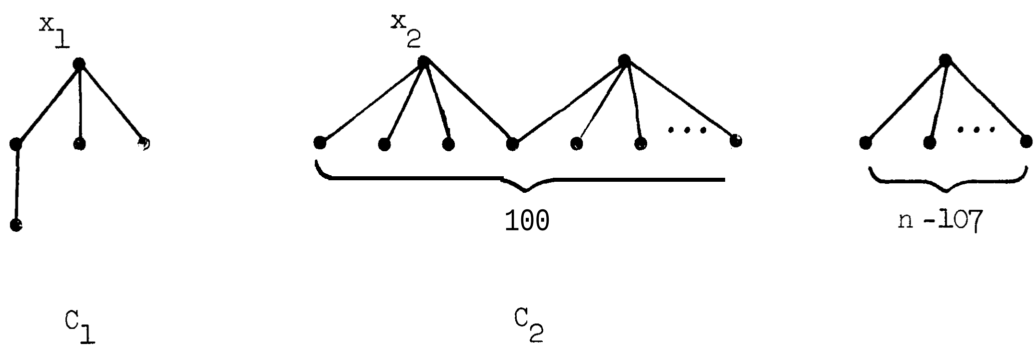

Figure 4. Difficulties caused by anomaly.

$\{x_j \mid x_j \le x_i$ (in P), and x_j is not less than any other maximal element in $P\}$.
We denote $|F(x_i)|$ by $f(x_i)$. Note that $F(x_i) \subseteq H(x_i)$, and the containment is proper if and only if x_i is anomalous. When x_i is anomalous, we call $f(x_i)$ the <u>anomaly degree</u> of x_i .

Let T be an algorithm that selects the k-th largest of n elements. For any internal node $v \in T$, the comparison at v , $x_i : x_j$, is said to be <u>anomalous of degree</u> m if either x_i or x_j has anomalous degree m .

<u>Theorem 4.1</u> (The Limited-Anomaly Theorem). Let T be an algorithm selecting the k-th largest of $x = \{x_1, x_2, \ldots, x_n\}$, and σ an input ordering. Then the number of anomalous comparisons of degree $\le m$ is at most $(2m+1)s_1(\sigma)$.

<u>Proof</u>. We assign a weight m+1-i to an anomalous element of degree i for $1 \le i \le m$, and a weight 0 to all other elements. Let E and E' be respectively the total weight of all elements before and after a comparison $x_i > x_j$. Then the following is true.

<u>Lemma 4.2</u>.

(A) $E' < E + 2m$.

(B) If $x_i > x_j$ is a first defeat, then $E' \le E$.

(c) If $x_i > x_j$ is a first defeat and an anomalous comparison of degree < m , then $E' < E$.

<u>Proof of Lemma 4.2</u>. It is easy to see that at most two elements will be assigned new weights after the comparison; namely, the two maximal elements y and z whose fiefdoms contain x_i and x_j respectively. Since the largest increase in weight for an element is from 0 to m , this proves (A).

To prove (B) note that $x_i > x_j$ is a first defeat implies $x_j = z$. After the comparison, z is no longer maximal, and $F(y) \leftarrow F(y) \cup F(z)$. We consider two cases according to whether z was anomalous of degree $< m$ before the comparison $x_i > x_j$.

Case (a). was anomalous of degree $\leq m$.

The decrease in z's weight is from $m+1 - f(z)$ to 0 while the maximum increase in y's weight is from 0 to $\max\{0, m+1 - (f(y)+ f(z))]$ $< m+1 - f(z)$. This means $E' < E$.

Case (b). was not anomalous of degree $\leq m$.

Then z's weight does not change; y's weight has two cases:

(b1) y was anomalous of degree $\leq m$. Then y's weight strictly decreases due to the strict increase in its anomaly degree.

(b2) Y was not anomalous of degree $< m$. Then y's weight remains 0 .

This proves (B). Statement (C) follows from the analysis of Case (a) and Case (bl) above. This proves Lemma 4.2. □

We will now complete the proof of Theorem 4.1. Statements (A) and (B) of Lemma 4.2 imply that the total increase in weight along path $S(\sigma)$ is bounded by $2ms_1(\sigma)$. Since the sum of weights of the elements is initially 0 and always non-negative by definition, the number of comparisons n_3 which fits statement (C) of Lemma 4.2 is at most $2ms_1(\sigma)$. The total number of comparisons along $S(0)$ that are anomalous of degree $< m$ is clearly at most $n_3 + s_1(\sigma)$, and is hence bounded by $(2m+1)s_1(\sigma)$. This proves Theorem 4.1. □

5. Proof of the Main Theorem.

5.1 Introduction.

We will prove the following result in this section.

Theorem 5.1. Let k, n be integers with $k \geq 2$ and $n \geq N_k = (8k)^{18k}$. Suppose T is an algorithm that selects the k-th largest of n elements, and σ any input ordering. Then $\alpha(\sigma) > k(\ln \ln n - \ln k - 6)$, if $s_1(\sigma) \leq n^{0.2}$.

As defined in Section 2, the quantities $\alpha(\sigma)$, $s_1(\sigma)$ depend on T. Also note that, for $n \geq N_k$, the following inequalities hold, as can be verified by elementary arguments.

$$\begin{cases} n^{0.1} \geq k \ln \ln n & (5.1) \\ n^{1/(6k)} \geq 21gn & (5.2) \\ n^{1/12} > k & (5.3) \end{cases}$$

We first demonstrate that Theorem 5.1 implies the Main Theorem. If there are more than $n! \times n^{-0.1}$ σ satisfying $s_1(\sigma) > n^{0.2}$, then (2.8) implies

$$\begin{aligned} \mathrm{COST}(T) &\geq n - k + \frac{1}{n!} n! \, n^{-0.1} n^{0.2} \\ &\geq n - k + k \ln \ln n , \end{aligned}$$

in view of (5.1). On the other hand if less than $n! \times n^{-0.1}$ of the σ's satisfy $s_1(\sigma) > n^{0.2}$, then (2.9) and Theorem 5.1 lead to

$$\begin{aligned} \mathrm{COST}(T) &\geq n - k + \frac{1}{2}\frac{1}{n!}(n! - n! \times n^{-0.1})k(\ln \ln n - \ln k - 6) \\ &> n + \frac{1}{2}k(\ln \ln n - \ln k - 6 - n^{-0.1} \ln \ln n - 2) . \end{aligned}$$

Again, using (5.1), we obtain

$$\text{COST}(T) \geq n + \frac{1}{2} k(\ln \ln n - \ln k - 9) .$$

Thus, the Main Theorem is true in both cases.

5.2 Some Results on Partial Orders.

Let P be a partial order on a set $X = \{x_1, x_2, \ldots, x_n\}$. Assume that all orderings on X consistent with P are equally likely. We are interested in bounds on the probability of some element x_i being greater than another element x_j (or all elements in some subset). For instance, if x_i is the unique maximal element in a component (in P) of size m , then the probability that x_i is the maximum of all n elements in X is clearly at least m/n , and it is also not difficult to show that $\Pr(x_i > x_j)$ is at least $m/(m+r-1)$, if x_j is a non-maximal element in a different component of size r , A generalization of these facts is given below in two lemmas.

<u>Lemma 5.2.</u> If x_i is a maximal element, then

$$\Pr(x_i \text{ is the largest element in } X) \geq \frac{f(x_i)}{n} \quad ,$$

<u>Lemma 5.3.</u> If x_i is a maximal element, and x_j a non-maximal element in a different component, then

$$\Pr(x_i > x_j) \geq \frac{f(x_i)}{f(x_i) + h(x_j) - 1}$$

Intuitively, the above lemmas must be true, since knowing that some elements in $F(x_i)$ are greater than some elements outside $F(x_i)$ should not lower the rank of x_1. However, the proofs are not trivial, and are given in [3] where related issues are studied.

Lemma 5.4. Suppose x_i is the unique element in a component c of size m, and x_j a non-maximal element in a different component C' of size Δ-m . Assume that $\Delta > 2k$. Define the quantity β to be $(\Pr(x_i > x_j) + \Pr(\max\{x_i, x_j\}$ is in the top k-l of X)) . Then

$$\beta \geq \min\{1-e^{-km/\Delta}, 1-e^{-tm/\Delta} + (\Delta/(2n))^t , 1 < t < k\} .$$

Proof. See Appendix--B. □

5.3 Lower Bounds on q&v) .

Let v be an internal node in the algorithm T . Suppose v compares x_i , x_j . We will give lower bounds on $q_\ell(v)$ in terms of component sizes such as $f(x_i)$, $h(x_j)$, etc. defined relative to P(v) .

Lemma 5.5. If x_i is a non-maximal element, then $q_1(v) > 1/h(x_i)$.

Proof. If x_j is also non-maximal, then $q_1(v) = 1$, else by Lemma 5.3, $q_1(v) = \Pr(x_j > x_i) > f(x_j)/(f(x_j)+h(x_i) -1) \geq 1/h(x_i)$. Cl

Lemma 5.6. If both x_i and x_j are maximal, then $q_2(v) \geq (f(x_i)+f(x_j))/n$.

Proof. The properties of x_i , x_j being the largest element in X are mutually exclusive. Hence $q_2(v) \geq \frac{f(x_i)}{n} + \frac{f(x_j)}{n}$ by Lemma 5.2. □

Lemma 5.7. If x_i is a maximal element and x_j a non-maximal element, then $q_1(v) > f(x_i)/(f(x_i)+h(x_j))$.

Proof. It follows directly from Lemma 5.3. □

Lemma 5.8. Suppose x_i is the unique maximal element in a component C , and x_j a non-maximal element in a different component. If $h(x_i) \le n^{1/3}$ and $h(x_i)+h(x_j) \ge n^{1-(1/6k)}$, then

$$q(v) \ge k\frac{h(x_i)}{h(x_j)} - 3k^2\frac{1}{n^{7/6}} .$$

Proof. Let $m = h(x_i)$, $m' = h(x_j)$ and $\Delta = m+m'$. Then by assumption

$$m \le n^{1/3} \quad \text{and} \quad \Delta \ge n^{1-(1/6k)} \tag{5.4}$$

Clearly $\Delta > 2k$. By Lemma 5.4, we need only show that

$$1-e^{-km/\Delta} \ge k\frac{m}{m'} - 3k^2\frac{1}{n^{7/6}} , \tag{5.5}$$

and

$$\min_{1<t<k}\left\{1-e^{-tm/\Delta}+\left(\frac{\Delta}{2n}\right)^t\right\} \ge k\frac{m}{m'} - 3k^2\frac{1}{n^{7/6}} . \tag{5.6}$$

As $e^{-x} \le 1 - x + \frac{1}{2}x^2$ for $x \ge 0$, we have

$$1-e^{-km/\Delta} \ge \frac{k}{\Delta}m - \frac{1}{2}\left(\frac{km}{\Delta}\right)^2$$

$$= k\frac{m}{m'} - k\frac{m^2}{\Delta m'} - \frac{1}{2}\left(\frac{km}{\Delta}\right)^2 . \tag{5.7}$$

Now, from (5.4),

$$\frac{m}{\Delta} \le n^{-\left(\frac{2}{3}-\frac{1}{6k}\right)} . \tag{5.8}$$

This implies $m/\Delta < 1/2$ and hence

$$m' > \frac{1}{2}\Delta \quad . \tag{5.9}$$

Using (5.8) and (5.9) in (5.7), we obtain

$$\begin{aligned} 1-e^{-km/\Delta} &\geq k\frac{m}{m'} - \left(2k + \frac{k^2}{2}\right)\left(\frac{m}{\Delta}\right) \\ &\geq k\frac{m}{m'} - 3k^2 n^{-\left(\frac{4}{3} - \frac{1}{3k}\right)} \\ &\geq k\frac{m}{m'} - 3k^2 n^{-\frac{7}{6}} \end{aligned} \tag{5.10}$$

This proves (5.5).

For $1 < t < k$,

$$\begin{aligned} 1-e^{-tm/\Delta} + \left(\frac{\Delta}{2n}\right)^t &\geq \left(\frac{\Delta}{2n}\right)^{k-1} \\ &\geq n^{-\frac{1}{6}+\frac{1}{6k}} \cdot 2^{-(k-1)} \\ &> 2kn^{-\left(\frac{2}{3}-\frac{1}{6k}\right)} \quad , \end{aligned} \tag{5.11}$$

where we have used (5.4) and the fact $n \geq N_k > k^2 4^k$. We now use (5X) and (5.9) to obtain

$$\begin{aligned} 1-e^{-tm/\Delta} + \left(\frac{\Delta}{2n}\right)^t &\geq 2k\frac{m}{\Delta} \\ &\geq k\frac{m}{m'} \quad . \end{aligned}$$

This implies (5.6) immediately. □

5.4 Completing the Proof.

As in Section 3.3, we construct a merge-tree $M(\sigma)$ corresponding to the merging process for σ, and assign $\varphi(u) = q(v)$ to each $u \in M(\sigma)_I$, It will be shown that, under the assumptions in Theorem 5.1,

$$\sum_{u \in M(\sigma)_I} \varphi(u) \geq k(\ln \ln n - \ln k - 6) . \tag{5.12}$$

This would prove Theorem 5.1, as

$$\begin{aligned} \alpha(\sigma) &= \sum_{v \in S(\sigma)} q(v) \\ &\geq \sum_{v \in S'(\sigma)} q(v) \\ &= \sum_{u \in M(\sigma)_I} \varphi(u) \quad . \end{aligned}$$

To prove (5.12), we first partition the set of nodes in M(a) into upper and lower parts, $U = (u \mid w(u) > n^{1/3} \}$ and $L = \{u \mid w(u) < n^{1/3}\}$. Let $V' = (u \mid u \in U, \ell son[u] \in L, rson[u] \in L\}$, $V'' = \{u \mid u \in L, father[u] \in U-V']$, and $V = V' \cup V''$. (These definitions are similar to those used in Appendix A, and properties P1 - P5 there remain true.)

We now partition V into seven disjoint parts $V_1, V_2, \ldots, V_7$. For each $u \in V$, we assign u to a unique V_i according to the following procedure, which halts as soon as u is assigned,

Procedure Decompose;

step 1: If there is some $u' \in D(u)$ where the joining comparison is not between two maximal elements, then assign u to V_1 .

[comment: If u is not assigned in step 1, then the joining comparison at u creates a component C(u) with a unique maximal element;

recall that C(u) consists of the x_i's that label the leaves in $D_L(u)$.]

step 2: If $u \in V'$, then assign u to V_2 .

[comment: If u has not been assigned after step 2, then u must be in V'' and father[u] exists.] .

step 3: If father[u] compares a non-maximal element in C(u) with any element, then assign u to V_3 .

step 4: If father[u] compares the maximal element of C(u) with another maximal element (in a different component), then

$$\text{assign u to } \begin{cases} V_4 & \text{if the comparison is anomalous of degree at most } \lceil n^{1/5} \rceil , \\ V_5 & \text{otherwise.} \end{cases}$$

step 5: If father[u] compares the maximal element of C(u) with some non-maximal element (in a different component), then

$$\text{assign u to } \begin{cases} V_6 & \text{if } w(\text{father}[u]) \le n^{1-\frac{1}{6k}} , \\ V_7 & \text{if } w(\text{father}[u]) > n^{1-\frac{1}{6k}} \end{cases}$$

end Decompose.

Let $W_i = \sum_{u \in V_i} w(u) \quad (1 \le i \le 7)$, and

$$A_i = \begin{cases} \sum_{u \in V_i} \sum_{u' \in D(u)} \varphi(u') & \text{if } i \in \{1,2,4\} , \\ \sum_{u \in V_i} \varphi(\text{father}[u]) & \text{if } i \in \{3,6,7\} , \\ \sum_{u \in V_i} \left(\sum_{u' \in D(u)} \varphi(u') + \varphi(\text{father}[u]) \right) & \text{if } i \in \{5\} . \end{cases}$$

In analogy with discussions in Appendix A, it is not difficult to see that V_7 is a cross section, and that

$$\sum_{1 \le i \le 7} W_i = n, \tag{5.13}$$

and

$$\sum_{u \in M(\sigma)_I} E(u) \ge \sum_{1 \le i \le 7} A_i \quad . \tag{5.14}$$

We will now find lower bounds to the A_i's in terms of the W_i's. We treat first A_i for $i \in \{1,3,6\}$, which are "costly" and thus efficient algorithms should not have large W_i for these values of i .

Lemma 5.9. $A_1 + A_3 + A_6 \ge (W_1 + W_2 + W_6) n^{-\left(1 - \frac{1}{6k}\right)}$.

Proof. For each $u \in V_1$, some $u' \in D(u)$ has a comparison involving a non-maximal element. Thus, by Lemma 5.5, $\sum_{u' \in D(u)} \varphi(u') \ge n^{-1/3}$. We have

$$A_1 \ge |V_1| \cdot n^{-1/3} \quad . \tag{5.15}$$

Similarly, by Lemma 5.5, we have

$$A_3 \ge |V_3| \cdot n^{-1/3} \quad . \tag{5.16}$$

As each $u \in V$ has $w(u) \le 2n^{1/3}$, we have for $i \in \{1,3\}$

$$|V_i| \ge \frac{1}{2} W_i \, n^{-1/3} \tag{5.17}$$

Formulas (5.15) - (5.17) lead to

$$\begin{aligned} A_i &\ge \frac{1}{2} W_i \cdot n^{-2/3} \\ &\ge W_i \cdot n^{-\left(1 - \frac{1}{6k}\right)} \quad , \quad \text{for } i \in \{1,3\} \ . \end{aligned} \tag{5.18}$$

For each $u \in V_6$, we apply Lemma 5.7 to father[u] and obtain

$$\varphi(\text{father}[u]) \geq \frac{w(u)}{w(\text{father}[u])}$$

$$\geq w(u) n^{-\left(1-\frac{1}{6k}\right)} .$$

Thus,

$$A_6 \geq \sum_{u \in V_6} w(u) n^{-\left(1-\frac{1}{6k}\right)}$$

$$= W_6 \, n^{-\left(1-\frac{1}{6k}\right)} . \qquad (5.19)$$

Combining (5.18) and (5.19), we obtain the lemma. $\square$

Lemma 5.10. $W_4 \leq 8n^{11/15}$.

Proof. By the Limited-Anomaly Theorem (Theorem 4.1),

$$|V_4| \leq (2\lceil n^{1/5}\rceil + 1) s_1(\sigma) \leq 8n^{0.4} ,$$

since $s_1(\sigma) \leq n^{0.2}$ by assumption. As each $u \in V_4$ has $w(u) < n^{1/3}$, we have

$$W_4 \leq |V_4| n^{1/3} \leq 8n^{11/15} . \quad \square$$

Lemma.5.11. $A_2 \geq \frac{W_2}{3n} \lg n - 1$.

Proof. Let $u \in V_2$. For each $u' \in D(u)$, $\varphi(u') \geq w(u')/n$ by Lemma 5.6, as the corresponding comparison is between normal maximal elements. This gives, by Lemma 3.2,

$$\sum_{u' \in D(u)} \varphi(u') \geq \frac{1}{n} \sum_{u' \in D(u)} w(u')$$

$$\geq \frac{1}{n} w(u)(\lg w(u) - 1) \quad .$$

As $w(u) \geq n^{1/3}$, we have

$$\sum_{u' \in D(u)} \varphi(u') \geq \frac{1}{n} w(u)\left(\frac{1}{3} \lg n - 1\right) \quad .$$

Therefore,

$$A_2 \geq \frac{1}{n} \sum_{u \in V_2} w(u)\left(\frac{1}{3} \lg n - 1\right)$$

$$\geq \frac{W_2}{3n} \lg n - 1 \quad . \quad \square$$

<u>Lemma 5.12.</u> $A_5 \geq \frac{W_5}{5n} \lg n - 1$.

<u>Proof.</u> If $|V_5| = 0$ then $W_5 = 0$ and the lemma is clearly true. We thus assume that $|V_5| > 0$. For each $u \in V_5$,

$$\sum_{u' \in D(u)} \varphi(u') \geq \frac{1}{n} w(u)(\lg w(u) - 1) \ ,$$

Thus, using Fact B in Section 3.3,

$$\sum_{u \in V_5} \sum_{u' \in D(u)} \varphi(u') \geq \frac{1}{n} \left(\sum_{u \in V_5} w(u) \lg w(u) - W_5 \right)$$

$$> \frac{1}{n} W_5 \lg \frac{W_5}{|V_5|} \tag{5.20}$$

Now, for each $u \in V_5$, let the comparison at father[u] be between x_i and x_j , where x_i is the maximal element of $C(u)$. By Lemma 5.6,

$$\varphi(\text{father}[u]) \geq \frac{f(x_i)+f(x_j)}{n}$$

$$\geq \begin{cases} \dfrac{w(\text{father}[u])}{n} \geq \dfrac{1}{n}\, n^{1/3} & \text{if } x_j \text{ is normal,} \\ \dfrac{1}{n} \lceil n^{1/5} \rceil & \text{if } x_j \text{ is anomalous.} \end{cases}$$

Thus,

$$\sum_{u \in V_5} \varphi(\text{father}[u]) \geq |V_5| n^{-4/5} \tag{5.21}$$

Formulas (5.20) and (5.21) lead to

$$A_5 \geq \frac{1}{n} W_5 \lg \frac{W_5}{|V_5|} + |V_5| n^{-4/5} - 1 \quad . \tag{5.22}$$

By standard minimization technique (e.g. see the proof of Fact E in Appendix A), (5.22) yields

$$A_5 \geq \frac{1}{n} W_5 \lg((\ln 2)\cdot n^{1/5}) + \frac{1}{n} W_5 \frac{1}{\ln 2} - 1 \quad .$$

The lemma follows, noting that $\lg \ln 2 + \frac{1}{\ln 2} > 0$. Cl

<u>Lemma 5.13</u>. $A_7 \geq k \ln \frac{n+1}{n - W_7 + 1} - 3$.

<u>Proof</u>. Let $u \in V_7$, we write $u' = \text{brother}[u]$. By Lemma 5.8 and (5.3), we have

$$\varphi(\text{father}[u]) \geq k \frac{w(u)}{w(u')} - 3k^2 \frac{1}{n^{7/6}}$$

$$\geq k \frac{w(u)}{w(u')} - 3 \frac{1}{n} \quad .$$

As V_7 is a cross section, we obtain from Lemma 3.3. that

$$A_7 \geq k \sum_{u \in V_7} \frac{w(u)}{w(u')} - 3$$

$$\geq k \ln \frac{n+1}{n - W_7 + 1} - 3 \qquad \square$$

We are now ready to prove (5.12), and hence Theorem 5.1. Using Lemmas 5.9, 5.11, 5.12, 5.13 and formula (5.14), we have

$$\sum_{u \in M(\sigma)_I} \varphi(u) \geq \sum_i A_i$$

$$\geq (W_1 + W_3 + W_6) n^{-\left(1 - \frac{1}{6k}\right)} + \frac{\lg n}{3n} W_2 + \frac{\lg n}{5n} W_5$$

$$+ k \ln \frac{n+1}{n+1 - W_7} - 5 \quad .$$

Making use of (5.2) and (5.13)

$$\sum_{u \in M(\sigma)_I} \varphi(u) \geq \frac{\lg n}{5n}(W_1 + W_2 + W_3 + W_5 + W_6) + k \ln \frac{n+1}{n - W_7 + 1} - 5$$

$$= (n - W_7) \frac{\lg n}{5n} + k \ln \frac{n+1}{n - W_7 + 1} - 5 - \frac{W_4}{5n} \lg n \; . \tag{5.23}$$

From Lemma 5.10 and (5.2),

$$\frac{W_4}{5n} \lg n \leq \frac{8}{5} \frac{n^{11/15}}{n} \lg n$$

$$\leq 2 \frac{\lg n}{n^{4/15}}$$

$$< 1 \; . \tag{5.24}$$

Therefore, (5.23) leads to

$$\sum_{u \in M(\sigma)_I} \varphi(u) \geq x \frac{\lg n}{5n} + k \ln \frac{n+1}{x+1} - 6 \quad ,$$

for some x, $0 \leq x \leq n$.

A standard minimization gives

$$\sum_{u \in M(\sigma)_I} \varphi(u) \geq k \ln \left(\frac{\lg n}{5k} \right) - 6$$

$$\geq k (\ln \ln n - \ln k - 6) ,$$

which is (5.12).

This completes the proof of the Main Theorem. $\square$

Appendix A: Proof of Lemma 3.5.

The lemma is clearly true when $n \le 8$. We shall thus assume that $n > 8$. Note that, in this range,

$$n^{1/3} > \max\left\{\frac{1}{3}\lg n\ ,\ \frac{1}{2}\ln\ln n\right\} \quad . \tag{A.1}$$

We say a node $u \in M_I$ to be of category 1 if $g(u) = \min\{w_1, w_2\}/(w_1 + w_2)$, and of category 2 otherwise. For a node u to be of category 1, we must have

$$\frac{\min\{w_1,w_2\}}{w_1+w_2} \le \frac{w_1+w_2}{n} \quad ,$$

implying

$$w(u) = w_1 + w_2 > \sqrt{n} \quad . \tag{A.2}$$

Let us divide the set of nodes of M into an upper part U and a lower part L according to whether or not $w(u) \ge n^{1/3}/2$. As $n > 8$, the root must be in U and all leaves are in L , Now consider the set V' of lowest nodes in U , i.e.,

$$V' = \{u \mid u \in U\ ,\ \ell son[u] \in L\ ,\ rson[u] \in L\} \ ,$$

and the set V'' defined by

$$V'' = \{u \mid u \in L,\ father[u] \in U\text{-}V'\} \ .$$

An alternative characterization of V'' is given by

$$V'' = \{u \mid u \in L,\ father[u] \in U,\ brother[u] \in U\}$$

Let $V = V' \cup V''$. The following simple properties are easy to check.

P1: V' and V'' are disjoint.

P2: Any two distinct nodes in V have no common descendants.

P3: Any two distinct nodes in V'' have distinct fathers; furthermore, the set $\{\text{father}[u] \mid u \in V''\}$ is disjoint from the union of descendants of nodes in V.

P4: V'' is a cross section of M.

P5: The family of sets $\{D_L(u) \mid u \in V\}$ forms a partition of the leaves of M.

We partition $V = V' \cup V''$ into V_i $(1 \leq i \leq 4)$ as follows. The set V_1 is simply V'. Sets V_2, V_3, V_4 are given by

$$V_2 = \{u \mid u \in V'', \text{father}[u] \text{ is of category } 2\},$$

$$V_3 = \{u \mid u \in V'', \text{father}[u] \text{ is of category } 1,\ w(\text{father}[u]) < n^{2/3}\},$$

$$V_4 = \{u \mid u \in V'', \text{father}[u] \text{ is of category } 1,\ w(\text{father}[u]) \geq n^{2/3}\}.$$

The definitions are illustrated in Figure 5.

Let $W_i = \sum_{u \in V_i} w(u)$ for $1 \leq i \leq 4$. Define

$$\begin{cases} A_1 = \sum_{u \in V_1} \sum_{u' \in D(u)} g(u') \\ A_2 = \sum_{u \in V_2} \left(\sum_{u' \in D(u)} g(u') + g(\text{father}[u]) \right) \\ A_i = \sum_{u \in V_i} g(\text{father}[u]) \qquad i = 3,4. \end{cases}$$

As an immediate consequence of property P5, we have

$$\sum_{1 \leq i \leq 4} W_i = n \quad . \tag{A.3}$$

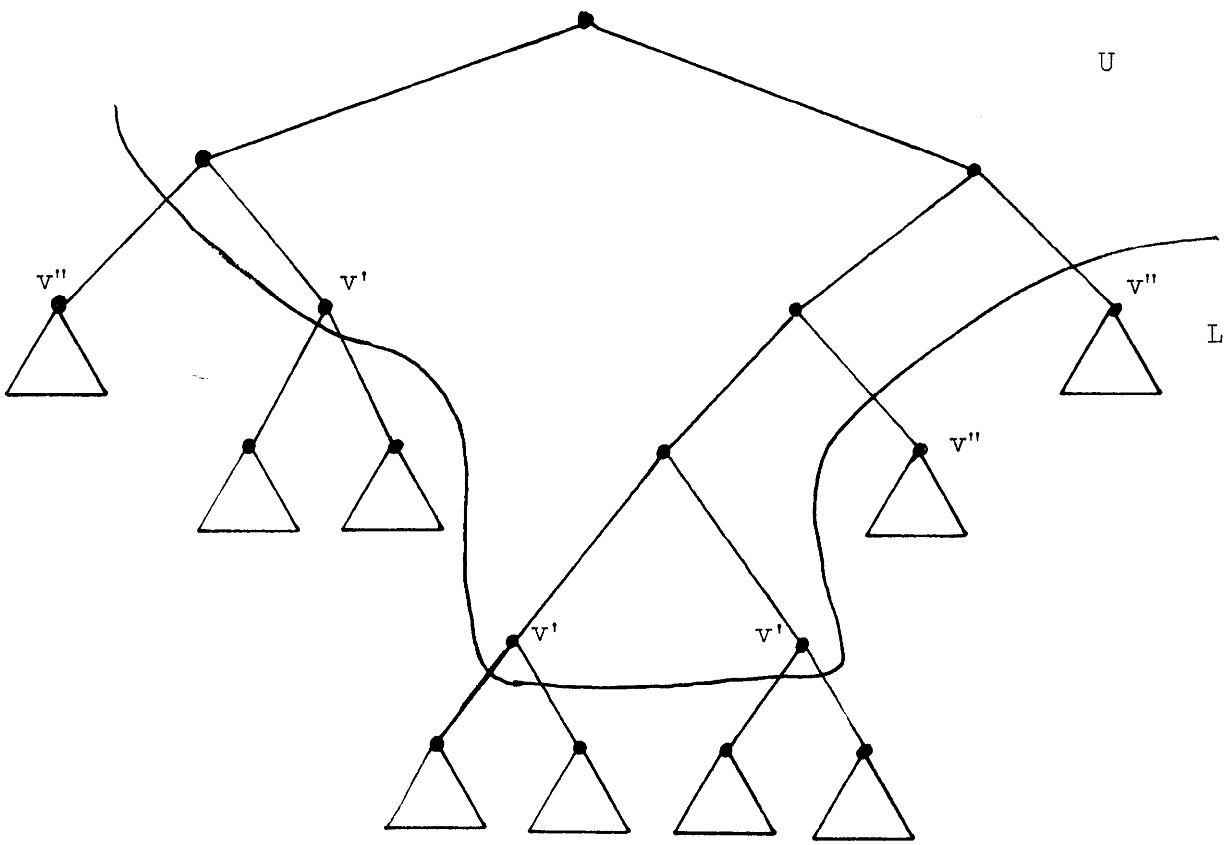

Figure 5. A schematic illustration of sets U, L, $V' = V_1$, $V'' = V_2 \cup V_3 \cup V_4$; nodes in V' , V'' are labeled as v' , v'' , respectively.

Now, from properties P1-P3, we have

$$\sum_{u \in M_I} \quad \geq \sum_{1 \leq i \leq 4} A_i \quad . \tag{A.4}$$

Our plan is to first derive lower bounds to A_i in terms of W_i, and then apply (A.4) to prove Lemma 3.5.

Fact C. If $w(u) < \sqrt{n}$, then $\sum_{u' \in D(u)} g(u') \geq \frac{1}{n} w(u)(\lg w(u) - 1)$.

Proof. We may assume that $u \in M_I$, as the assertion is clearly true when u is a leaf. Now each $u' \in D(u)$ must be of category 2 $(w(u') < \sqrt{n})$, and hence $g(u') = w(u')/n$. Using Lemma 3.2, we have

$$\sum_{u' \in D(u)} g(u') = \frac{1}{n} \sum_{u' \in D(u)} w(u')$$

$$\geq \frac{1}{n} w(u)(\lg w(u) - 1) \quad . \quad \square$$

Fact D. $A_1 \geq \frac{W_1}{3n} \lg n - 2$.

Proof. Each $u \in V_1$ satisfies $w(u) < 2(n^{1/3}/2) \leq \sqrt{n}$, and hence from Fact C,

$$A_1 = \sum_{u \in V_1} \sum_{u' \in D(u)} g(u')$$

$$> \sum_{u \in V_1} \frac{1}{n} w(u)(\lg w(u) - 1)$$

$$> \frac{1}{n} \sum_{u \in V_1} w(u) \lg w(u) - 1 \; .$$

As $w(u) \geq n^{1/3}/2$ (since $u \in U$), we have

$$A_1 \geq \frac{1}{n} \sum_{u \in V_1} w(u)\left(\frac{1}{3} \lg n - 1\right) - 1$$

$$\geq \frac{W_1}{3n} \lg n - 2 \quad . \quad \square$$

<u>Fact</u> E. $A_2 \geq \dfrac{W_2}{3n} \lg n - 3$.

<u>Proof.</u> The statement is obviously true when $|V_2| = 0$. We shall. thus assume that $|V_2| > 0$. For each $u \in V_2$, $g(father[u]) = w(father[u])/n \geq 1/(2n^{2/3})$, since father[u] is of category 2 and is in U . Making use of Fact C, we have

$$A_2 = \sum_{u \in V_2} \sum_{u' \in D(u)} g(u') + \sum_{u \in V_2} g(father[u])$$

$$> \sum_{u \in V_2} \frac{w(u)}{n} (\lg w(u) - 1) + |V_2| \frac{1}{2n^{2/3}} \quad .$$

We now use Fact B to obtain

$$A_2 \geq \frac{W_2}{n} \lg \frac{W_2}{|V_2|} - 1 + \frac{|V_2|}{2n^{2/3}} \tag{A.5}$$

The right hand side expression $d(|V_2|)$ achieves its absolute minimum over $|V_2| \in [0,\infty)$ at $|V_2| = 2W_2/(n^{1/3} \ln 2)$, where

$$d(|V_1|) = \frac{W_2}{n} \lg\left(\frac{\ln 2}{2} n^{1/3}\right) - 1 + \frac{1}{\ln 2} \frac{W_2}{n}$$

$$\geq \frac{W_2}{3n} \lg n - 3 \quad .$$

Thus, formula (A.5) implies

$$A_2 \geq \frac{W_2}{3n} \lg n - 3 \quad , \tag{A.6}$$

proving Fact E. □

The derivation of (A.6) from (A.5) is a standard argument, and similar derivations will henceforth be referred to as "by standard minimization technique" with details omitted.

For each $u \in V_3 \cup V_4$, $w(\text{brother}[u]) \geq n^{1/3}/2 > w(u)$, and $\text{father}[u]$ is of category 1. Thus,

$$g(\text{father}[u]) = \frac{w(u)}{w(\text{father}[u])} \quad . \tag{A.7}$$

Fact F. $A_3 \geq \frac{W_3}{n^{2/3}}$.

Proof. For each $u \in V_3$, $w(\text{father}[u]) < n^{2/3}$. Using (A.7), we have

$$\begin{aligned} A_3 &= \sum_{u \in V_3} g(\text{father}[u]) \\ &= \sum_{u \in V_3} \frac{w(u)}{w(\text{father}[u])} \\ &\geq \frac{W_3}{n^{2/3}} \quad . \quad \square \end{aligned}$$

Fact G. $A_4 \geq \left(1 - \frac{1}{2n^{1/3}}\right) \ln \frac{n+1}{n - W_4 + 1}$.

Proof. For each $u \in V_4$, $w(u) < n^{1/3}/2$ and $w(\text{father}[u]) \geq n^{2/3}$. Using (A.7), we have

$$g(\text{father}[u]) = \frac{w(u)}{w(\text{father}[u])}$$

$$= \frac{w(u)}{w(\text{brother}[u])}\left(1 - \frac{w(u)}{w(\text{father}[u])}\right)$$

$$\geq \frac{w(u)}{w(\text{brother}[u])}\left(1 - \frac{1}{2n^{1/3}}\right)$$

Thus,

$$A_4 = \sum_{u \in V_4} g(\text{father}[u])$$

$$\geq \left(1 - \frac{1}{2n^{1/3}}\right) \sum_{u \in V_4} \frac{w(u)}{w(\text{brother}[u])} \quad . \tag{A.8}$$

As V'' is a cross section of M by property P4, so is V_4. Fact G then follows from (A.8) and Lemma 3.3. ❑

We will now finish the proof of Lemma 3.5. Using Facts D - G, we obtain from (A.4)

$$\sum_{u \in M_I} g(u) \geq \frac{W_1 + W_2}{3n} \lg n + \frac{W_3}{n^{2/3}} + \left(1 - \frac{1}{2n^{1/3}}\right) \ln \frac{n+1}{n - W_4 + 1} - 5 \quad .$$

Using (A.1) and (A.3), we obtain then

$$\sum_{u \in M_I} g(u) \geq \frac{W_1 + W_2 + W_3}{3n} \lg n + \left(1 - \frac{1}{e\ 1\ n}\right) \ln \frac{n+1}{W_1 + W_2 + W_3 + 1} - 5$$

$$\geq \left(1 - \frac{1}{2n^{1/3}}\right)\left(\frac{x}{3n} \lg n + \ln \frac{n+1}{x+1}\right) - 5 \quad , \tag{A.9}$$

where $x = W_1 + W_2 + W_3$.

By standard minimization technique, we obtain from (A.9)

$$\sum_{u \in M_I} g(u) \geq \left(1 - \frac{1}{2n^{1/3}}\right)(\ln \ln n - 1) - 5$$

$$> \ln \ln n - 7 \quad ,$$

where (A.1) was used in the last step, This proves Lemma 3.5. $\square$

Appendix B: Proof of Lemma 5.4.

Let $\beta(t)$ be the quantity β when the component C' has been sorted and x_j is the t-th largest in it, Then, denoting by $p(t)$ the probability that x_j is the t-th largest in C' under partial order P , we have with $m' = \Delta - m$,

$$\beta = \sum_{1 \le t \le m'} p(t)\beta(t) \quad .$$

As x_j is not a maximal element, $p(1) = 0$. Therefore, the lemma would follow, if we can show that for all $1 < t \le m'$,

$$\beta(t) \ge \min \left\{ 1-e^{-km/\Delta} , 1-e^{-t'm/\Delta} + \left(\frac{\Delta}{2n}\right)^{t'} \quad \text{for } 1 < t' < k \right\} . \qquad (B.1)$$

Let $\beta(t) = a_1 + a_2$, where

$$\begin{cases} a_1 = \text{probability that } x_i > x_j , \\ a_2 = \text{probability that } \max\{x_i, x_j\} \text{ is in the top k-l.} \end{cases}$$

Clearly,

$$\begin{aligned} a_1 &= 1 - (\text{probability } x_i < x_j) \\ &= 1 - \frac{\binom{\Delta-t}{m}}{\binom{\Delta}{m}} \\ &= 1 - \left(1 - \frac{t}{\Delta}\right)\left(1 - \frac{t}{\Delta-1}\right) \cdot ** \left(1 - \frac{t}{\Delta-m+1}\right) \\ &\_ \; 1 - \left(1 - \frac{t}{\Delta}\right)^m . \end{aligned}$$

But,

$$\left(1 - \frac{t}{\Delta}\right)^m = e^{m \ln(1 - t/\Delta)}$$

$$\leq e^{m(-t/\Delta)} .$$

Thus,

$$a_1 \geq 1-e^{-tm/\Delta} \quad \text{for} \quad 1 < t < m' . \tag{B.2}$$

Formula (B.2) proves (B.1) for the case $k \leq t \leq m'$. We shall now restrict our attention to the case $1 < t < k' = \min\{k,m'+1\}$. In this range,

$$a_2 = \Pr(\max\{x_i, x_j\} \text{ is in the top } k-1 \text{ of } X)$$

$$\geq \Pr(\text{the } t\text{-th largest element in } C \cup C' \text{ is in the top } k-1 \text{ of } X)$$

$$= \sum_{t \leq \ell < k} \Pr(\text{the } t\text{-th largest element in } C \cup C' \text{ is the } \varphi\text{-th largest in } X)$$

$$= \sum_{t \leq \ell < k} \frac{\binom{n-\ell}{\Delta-t}\binom{\ell-1}{t-1}}{\binom{n}{\Delta}} .$$

Taking only the term $\ell = t$ and using the assumption $A > 2k$, we obtain

$$a_2 \geq \frac{\Delta}{n}\,\frac{A-1}{n-1}\cdots\frac{\Delta-t+1}{n-t+1}$$

$$\geq \left(\frac{\Delta-k}{n}\right)^t$$

$$\geq \left(\frac{\Delta}{2n}\right)^t , \qquad \text{when } 1 < t < k' . \tag{B.3}$$

From (B.2) and (B.3), we see that for $1 < t < k'$

$$\beta(t) = a_1 + a_2$$

$$\geq 1 - e^{-tm/\Delta} + \left(\frac{\Delta}{2n}\right)^t .$$

Thus, (B.1) is also true in this case.

This completes the proof of Lemma 5.4. $\square$

References

[1] R. W. Floyd and R. L. Rivest, "Expected time bounds for selection," Communications ACM 18(1975), 165-172.

[2] F. Fussenegger and H. N. Gabow, "Using comparison trees to derive lower bounds for selection problems," Proc. 17-th IEEE Symp. on Foundations of Computer Science (1976), 178-182.

[3] R. L. Graham, A. C. Yao, and F. F. Yao, "Some monotonicity properties in partial. orders," to appear.

[4] L. Hyafil, "Bounds for selection," SIAM J. on Computing 5(1976), 109-144.

[5] D. G. Kirkpatrick, "Topics in the complexity of combinatorial algorithms," Computer Science Department Technical Report TR 74(1974), University of Toronto.

[6] D. E. Knuth, "Mathematical analysis of algorithms," Information Processing 71 (Proceedings of the 1971 IFIP Congress), North-Holland, Amsterdam, 1972, 19-27.

[7] D. E. Knuth, The Art of Computer Programming, Vol. 1, Fundamental Algorithms, Addison-Wesley, Reading, Mass., 1968.

[8] D. E. Knuth, The Art of Computer Programming, Vol. 3, Sorting and Searching, Addison-Wesley, Reading, Mass., 1st printing, 1973.

[9] D. W. Matula, "Selecting the t-th best in average n+ O(log log n) comparisons," TR 73-9 (1973), Washington University, St. Louis.

[10] V. R. Pratt and F. F. Yao, "On lower bounds for computing the i-th largest element," Proc. 14-th IEEE Symp. on Switching and Automata Theory (1973), 70-81.

[11] A. Schönhage, M. Paterson, and N. Pippenger, "Finding the median," JCSS 13 (1976), 184-199.

[12] C. K. Yap, "New lower bounds for median and related problems," Computer Science Department Research Report No, 79(1976), Yale University.