A COUNTING APPROACH TO LOWER BOUNDS FOR SELECTION PROBLEMS

by

Frank Fussenegger*
Martin Marietta Data Systems
Denver, Colorado

Harold N. Gabow
Department of Computer Science
University of Colorado at Boulder
Boulder, Colorado 80309

#CU-CS-107-77

August, 1977
Rev. June, 1978

*Mailing Address:
G4250, Martin Marietta Data Systems,
P. O. Box 179, Denver, Colorado 80201

Abstract

Lower bounds are derived on the number of comparisons to solve several well-known selection problems. Among the problems are: finding the t largest elements of a given set, in order (W_t); finding the s smallest and t largest elements, in order $(W_{s,t})$; and finding the t^{th} largest element (V_t). The results follow from bounds for more general selection problems, where an arbitrary partial order is given. The bounds for W_t and V_t generalize to the case where comparisons between linear functions of the input are allowed.

The approach is to show a comparison tree for a selection problem contains a number of trees for smaller problems, thus establishing a lower bound on the number of leaves. An equivalent approach uses an adversary, based on a numerical "chaos" function that measures the number of unknown relations.

Key Words

selection problems, lower bounds, comparisons, comparison trees.

1. Introduction

The discovery of a linear median-finding algorithm [BFPRT] sparked an interest in determining the exact complexity of selection problems [H,Kir,PY,SPP,Yap]. This paper derives lower bounds for a number of well-known selection problems, using one basic technique.

To define the problems, consider a linear ordered set, e.g., $\{1,\ldots,n\}$. A permutation of the set, $(e_1,\ldots,e_n)$, is given. The problem is to find certain elements, by comparing elements e_i and e_j. For example, the W_t problem is to find the t largest elements, in order. (Thus for $t = n$, the problem is to sort the input permutation.) Other problems are:

V_t -- Find the t^{th} largest element. (For $t = \lceil \frac{n}{2} \rceil$, the problem is to find the median.)

U_t -- Find the t largest elements, as a set. (For $t = \lfloor \frac{kn}{100} \rfloor$, the problem is to find the elements in the upper k percentiles.)

$W_{s,t}$ -- Find the s smallest and t largest elements, in order. (For $s = t = 1$, the problem is to find the maximum and the minimum.)

We investigate the worst-case number of comparisons needed to solve selection problems. For this, the function $W_t(n)$ is defined as the number of comparisons needed to find the t largest elements; similar functions are used for the other selection problems. (Occasionally, we use $W_t(n)$ to refer to the W_t problem on n elements; no confusion results from this.)

We also investigate two variations of the basic selection problems. In the first variation, in addition to the input permutation $(e_1,\ldots,e_n)$, we are given a partial order P of known relations on elements e_i. The functions $W_t(P)$, etc., denote the number of comparisons to solve these problems.

In the second variation, the input is a vector of real numbers $(e_1,\ldots,e_n)$; comparisons between arbitrary linear functions of the input are allowed. The functions $W_t^*(n)$, etc., denote the number of comparisons to solve these problems.

We present lower bounds that are currently the best known for the following functions: $W_t(n)$, $W_t(P)$, $W_t^*(n)$, $W_{s,t}(n)$, $W_{s,t}(P)$, $V_t(P)$, $V_t^*(n)$, $U_t(P)$, $U_t^*(n)$. The basic method is to show that a comparison tree for a selection problem (e.g., $W_t(n)$) contains a number of comparison trees for a simpler selection problem (e.g., $W_1(n-t+1)$). An alternate formulation uses an adversary.

2. Element-to-Element Comparions

This section derives a lower bound for $W_t(n)$, by means of comparison trees. An equivalent analysis based on an adversary is given. The adversary is then applied to the n and P versions of V_t, U_t, and $W_{s,t}$.

An algorithm for a selection problem can be represented by a labelled binary tree called a comparison tree. Each interior node has a label, $i:j$; this represents a comparison made by the algorithm between elements e_i and e_j. If $e_i < e_j$, the algorithm's subsequent comparisons are represented in the node's left

subtree; otherwise, $e_i > e_j$ (since the input elements are assumed distinct), and subsequent comparisons are represented in the node's right subtree. Each leaf node is labelled with the appropriate answer to the problem; e.g., for W_t, a leaf lists the indices of the t largest elements, in order. Figure 1 shows a comparison tree for $W_2(4)$. The height of a comparison tree is the (worst case) number of comparisons used by the algorithm.

A node in a comparison tree is feasible if some imput permutation leads to it. Throughout the discussion we assume comparison trees contain only feasible nodes. This does not restrict the generality of the results: It is easy to see infeasible nodes can be eliminated without increasing the height of the tree. So a lower bound on the height of a tree with all nodes feasible holds in general.

We first examine comparison trees for W_1. These trees find the maximum element.

Lemma 1: A comparison tree for $W_1(n)$ has at least 2^{n-1} leaves.

Proof: In a sequence of comparisons that determines the maximum element, every non-maximum element compares low at least once. Thus at least $n-1$ comparisons are needed. So any path in the comparison tree from the root to a leaf contains at least $n-1$ interior nodes. If all paths to leaves contain exactly $n-1$ interior nodes, there are 2^{n-1} leaves; otherwise there are more. □

Now we analyze W_t, by showing a comparison tree for $W_t(n)$ "contains" a large number of comparison trees for $W_1(n-t+1)$.

Theorem 1: For any t, n, where $1 \le t \le n$,

$$W_t(n) \ge n - t + \lceil \lg n(n-1)\ldots(n-t+2) \rceil. \ *$$

Proof: Let T be a comparison tree for $W_t(n)$. A leaf in T lists the indices of the t largest elements. We show every sequence of $t-1$ indices is listed in at least 2^{n-t} leaves as the indices of the $t-1$ largest elements. This implies the Theorem. For there are $n(n-1)\cdot\ldots\cdot(n-t+2)$ sequences of indices, whence at least $n(n-1)\cdot\ldots\cdot(n-t+2)2^{n-t}$ leaves. Any binary tree with ℓ leaves has height at least $\lceil \lg \ell \rceil$. This gives a lower bound on the height of T, which is the desired bound on $W_t(n)$.

So fix a sequence of $t-1$ indices. For ease of notation, let the sequence be $n, n-1, \ldots, n-t+2$. Imagine these are the indices of the $t-1$ largest elements (in order), i.e., for any i, j, $n-t+2 \le i \le n$, $1 \le j \le n$, if $i > j$ then $e_i > e_j$. Prune T to a tree S, using these relations. Specifically, suppose a node x compares some e_i, $n-t+2 \le i \le n$. One of the above relations gives the outcome of the comparison. Replace x by the son for the appropriate outcome. (This eliminates x and one of its subtrees.) S results from making all such replacements. For example, Figure 2 shows S if T is the tree of Figure 1, and the sequence of largest indices is 4.

Now we examine S. Let $\underset{\sim}{e}$ be a permutation of $1,\ldots,n-t+1$, $\underset{\sim}{e} = (e_1,\ldots,e_{n-t+1})$. Form $\underset{\sim}{f}$, a permutation of $1,\ldots,n$, by making $n,\ldots,n-t+2$ the indices of the $t-1$ largest elements, i.e.,

---

* Throughout this paper, "lg" denotes logarithm to the base 2.

$\underset{\sim}{f} = (e_1,\ldots,e_{n-t+1}, n-t+2,\ldots,n)$. It is easy to see $\underset{\sim}{e}$ leads to the same leaf in S as $\underset{\sim}{f}$ leads to in T.

This implies a leaf in S lists $n,\ldots,n-t+2$ as the $t-1$ largest indices. So the Theorem follows if we show S has at least 2^{n-t} leaves. A leaf lists the index of the t^{th} largest element of $\underset{\sim}{f}$. This is the index of the largest element of $\underset{\sim}{e}$. So S (with a minor modification of labels) is a comparison tree that finds the maximum of $n-t+1$ elements. Now Lemma 1 shows S has 2^{n-t} leaves. □

Let $w_t(n)$ be the bound of Theorem 1, $w_t(n) = n - t + \lceil \lg n(n-1)\ldots(n-t+2) \rceil$. Now we examine its accuracy.

Corollary 1: For any t, n, $2 \le t \le n$,

(a) $w_t(n) \le W_t(n) \le w_t(n) + t - 2$;

(b) for $n \ge 2^{\lceil \lg n \rceil} - \frac{1}{t-1} + t - 2$, $W_t(n) = w_t(n)$.

Proof: The algorithm of repeated tree selection [Kis;Kn, p.212] shows $W_t(n) \le n - t + \sum_{i=n-t+2}^{n} \lceil \lg i \rceil$. To get $w_t(n)$ from this upper bound, we just change $\sum_{i=n-t+2}^{n} \lceil \lg i \rceil$ to $\left\lceil \sum_{i=n-t+2}^{n} \lg i \right\rceil$. For (a), note the definition of $\lceil\ \rceil$ guarantees this change decreases the value at most $t-2$. For (b), note if the condition on n is true, there is no change in value. □

Part (a) shows $w_t(n)$ is exact for $t=2$; it is obviously exact for $t=1$. (This result was obtained previously by Kislitsyn [Kis].) The hypothesis of (b) holds when n is a sufficiently large power of 2. Thus for any t, $w_t(n)$ is exact for infinitely many

values of n. However it is not always exact. For example, $w_n(n)$ is the information-theoretic bound on sorting, and so is off by 1 for $n=12$ [W;Kn,pp.188-193]. Also it can be shown $W_3(6)=9$, whereas $w_3(6)=8$ [M]. Nonetheless, the gap of $t-2$ given by part (a) does improve the previous gap of over $(t-1)^2$ [PY].

Now we express the derivation of Theorem 1 in terms of an adversary, and then apply the adversary to other selection problems. We could obtain the same results by arguments similar to Theorem 1 [FG]. We use the adversary for convenience and because of possible improvements it suggests.

The adversary is based on a function called "chaos" that measures the relevant information unknown to the algorithm. We represent the information known to the algorithm by a partial order P on elements $e_1,\ldots,e_n$. P contains all relations $e_i > e_j$ that have been indicated by comparisons. For example, Figure 3 is the Hasse diagram of P when the leftmost interior node of Figure 1 (labelled 4:2) is reached. In Figure 3, a downward edge from i to j denotes $e_i > e_j$.

An element e is <u>maximal</u> in P if there is no element f, $f > e$. The function <u>$m(P)$</u> gives the number of maximal elements in P. In Figure 3, $m(P) = 1$.

Next we define the orders that are "consistent" with P. Let <u>a</u> be a 1-1 assignment, defined on some subset of $\{e_1,\ldots,e_n\}$ with values in $\{1,\ldots,n\}$. Call <u>a</u> <u>consistent with P</u> if it can be extended to a 1-1 assignment $\bar{a}$, defined on all of $\{e_1,\ldots,e_n\}$ with values $\{1,\ldots,n\}$, such that $e > f$ implies $\bar{a}(e) > \bar{a}(f)$. The set <u>$A_k(P)$</u> contains all consistent assignments defined on k elements of $\{e_1,\ldots,e_n\}$, with values $\{n,n-1,\ldots,n-k+1\}$. Also, let <u>$a\cdot P$</u> denote

the partial order formed by removing all elements that $\underline{a}$ is defined on. In Figure 3, if $a(e_1) = 4$, $a(e_4) = 3$ and $a(e_2) = 2$, then $a \in A_3(P)$, and $a \cdot P$ contains the single element e_3.

Now define the chaos in P as

$$C_t(P) = \sum_{a \in A_{t-1}(P)} 2^{m(a \cdot P)-1} ,$$

where t is an integer in $1 \leq t \leq n$. The term $2^{m(a \cdot P)-1}$ is the number of leaves in a comparison tree that finds the t^{th} largest element, given P and the $t-1$ largest elements specified by $\underline{a}$. Hence $C_t(P)$ corresponds to our bound on the number of leaves in a comparison tree. For example, in a partial order P_o containing no relations, the chaos is maximum; each of $n(n-1)\ldots(n-t+2)$ possible assignments is consistent, so

$$C_t(P_o) = n(n-1)\ldots(n-t+2)2^{n-t} . \tag{1}$$

The chaos is minimum in a partial order P_f where the t largest elements are known (see Figure 4); here

$$C_t(P_f) = 1. \tag{2}$$

By convention if P is not a partial order, $C_t(P) = 0$.

A basic fact is that total chaos does not decrease when a comparison is made. More precisely, suppose elements e and f are compared. Let P be the partial order before the comparison. Define $P_>$ as the order P with the relation $e > f$ added. (Note $P_>$ is a partial order, unless $e > f$ introduces a cycle.) Define $P_<$ similarly.

Lemma 2: $C_t(P) \le C_t(P_>) + C_t(P_<)$.

Proof: If only one of $P_>$, $P_<$ is a partial order, say $P_>$, then it is easy to see $C_t(P_>) = C_t(P)$ and the Lemma holds. So assume both $P_>$ and $P_<$ are partial orders. Consider any $a \in A_{t-1}(P)$. We will show

(3) $$2^{m(a\cdot P)} \le 2^{m(a\cdot P_>)} + 2^{m(a\cdot P_<)} .$$

(If $\underline{a}$ is not consistent with $P_>$ then $2^{m(a\cdot P_>)}$ is 0, by convention; similarly for $P_<$). Inequality (3) suffices to prove the Lemma, since multiplying by 2^{-1} and summing over all $a \in A_{t-1}(P)$ gives the desired inequality.

To show (3), first suppose $\underline{a}$ is consistent with only one outcome, say $e > f$. Then $a(e)$ is defined, e is not in $a\cdot P$, $a\cdot P_> = a\cdot P$, and (3) follows.

Next suppose $\underline{a}$ is consistent with both outcomes. The relation $e > f$ can decrease the number of maximal elements in $a\cdot P$ by at most 1, i.e., $m(a\cdot P_>) \ge m(a\cdot P)-1$. Similarly $m(a\cdot P_<) \ge m(a\cdot P)-1$. These two inequalities imply (3). □

This non-decreasing property of chaos is the basis for our adversary. (An adversary, also called an oracle, is a rule specifying the outcome of a comparison. It tries to force the algorithm to make a large number of comparisons.) Suppose the partial order P represents the relations known, and the algorithm compares elements e and f. The adversary specifies the outcome according to the rule:

(*) If $C_t(P_>) > C_t(P_<)$, then $e > f$. Otherwise, $e < f$.

If (*) chooses $e > f$, the $P_>$ is the new partial order, and by Lemma 2, $C_t(P) \le 2C_t(P_>)$; similarly if $e < f$. A similar statement holds if

(*) chooses $e < f$. Thus if P_i is the partial order after the algorithm makes i comparisons, $C_t(P_0) \le 2^i C_t(P_i)$. Hence if f is the number of comparisons to solve a problem using the adversary,

$$f \ge \lceil \lg (C_t(P_0)/C_t(P_f)) \rceil . \tag{4}$$

Note (4) is a property of the adversary, and holds for any selection problem and algorithm.

Now it is easy to analyze selection problems where a partial order P is given.

Theorem 2: For any P, t, n, where P is a partial order on n elements and $1 \le t \le n$,

(a) $W_t(P) \ge \lceil \lg C_t(P) \rceil$;

(b) $V_t(P) \ge \lceil \lg(C_t(P)/(t-1)!) \rceil$;

(c) $U_t(P) \ge \lceil \lg(C_t(P)/t!) \rceil$.

Proof: Applying the adversary to each problem, (4) holds with $P_0 = P$. It is only necessary to calculate $C_t(P_f)$ and substitute.

For $W_t(P)$, the final partial order P_f is shown in Figure 4, and (2) holds. Thus (4) implies part (a).

For $V_t(P)$, P_f is shown in Figure 5(a): the t^{th} largest element e is known to be less than $t - 1$ elements and greater than $n - t$ elements. For $a \in A_{t-1}(P_f)$, $a \cdot P_f$ has only one maximal element, e. There are at most $(t-1)!$ possibilites for a (possibly fewer, since some permutations of the $t - 1$ elements larger than e may be inconsistent with P_f). Thus $C_t(P_f) \le (t-1)!$. Substituting in (4) gives part (b).

For $U_t(P)$, P_f is shown in Figure 5(b): t elements are known greater than the remaining $n - t$. If $\underline{a} \in A_{t-1}(P_f)$, $\underline{a} \cdot P_f$ contains exactly one maximal element, one of the t largest elements. There are at most $t!$ possibilities for $\underline{a}$. Thus $C_t(P_f) \leq t!$, and part (c) follows. □

Most previous bounds on selection functions do not apply to arbitrary partial orders. However Floyd [Kn,p.219,ex.6] has completely determined $W_2(P)$, as follows: If P has m maximal elements, where the j^{th} maximal element is immediately above ℓ_j elements in the Hasse diagram, then $W_2(P) = m - 2 + \lceil \lg \sum_{j=1}^{m} 2^{\ell_j} \rceil$. Theorem 2 gives an alternate proof of the lower bound part of this equality: There are m consistent assignments, the j^{th} one having $m - 1 + \ell_j$ maximal elements. Thus $C_t(P) = \sum_{j=1}^{m} 2^{m-2+\ell_j}$, and Floyd's lower bound follows.

Now we specialize Theorem 2 to the standard W_t, V_t, and U_t problems. When P contains no relations, the chaos is given by (1). Substituting in Theorem 2 gives Theorem 1 for $W_t(n)$, and also the following bounds.

Corollary 2: For any n, t, where $1 \leq t \leq n$,

$$V_t(n) \geq n - t + \lceil \lg \binom{n}{t-1} \rceil ;$$

$$U_t(n) \geq n - t + \lceil \lg(\binom{n}{t-1}/t) \rceil \ . \qquad \square$$

$V_t(n)$ has been extensively studied. Our bound is stronger than several others over large regions ([H,PY]). However it can be $\frac{3}{2}\,t$ smaller than Kirkpatrick's bound [Kir]. For the interesting case of

the median ($t=\lceil n/2 \rceil$), our bound is approximately $\frac{3}{2}n$, compared with Yap's bound of $\frac{11}{6}n$ [Yap]. Similar remarks apply to $U_t(n)$.

The above results are special cases of a bound for the General Partition Problem $P(i_1,\ldots,i_k)$. This problem was introduced by Yap [Yap]. We are given a set of n elements, where $n = \sum_{j=1}^{k} i_j$, and must find the set of i_1 largest elements, i_2 next largest elements, ..., i_k smallest elements. So for example the $W_t(n)$ problem is $P(1,\ldots,1,n-t)$, where there are t 1's. Using the chaos function C_{n-i_k} in (4) gives

$$(5) \qquad P(i_1,\ldots,i_k) \geq i_k + \lceil \lg \binom{n}{i_1,i_2,\ldots,i_k} / (i_k+1) \rceil .$$

A similar bound holds for the General Partition Problem when we are given an inital partial order.

This bound is superior to one derived from the information-theoretic bound on sorting [DM]. Let $S(n)$ be the number of comparisons needed to sort n elements, so $S(n) \geq \lceil \lg n! \rceil$. Since an algorithm for P can be used as the first step in a sort of n elements, we have

$$P(i_1,\ldots,i_k) \geq \lceil \lg n! \rceil - \sum_{j=1}^{k} S(i_j).$$

Substituting an upper bound for $S(i_j)$ gives the desired lower bound on P. However, since $S(i_j) \geq \lceil \lg i_j! \rceil$, it is easy to see such a lower bound is at most $\lceil \lg \binom{n}{i_1,\ldots,i_k} \rceil$, which is no larger than the right-hand side of (5). (The bound (12) below is often better than (5).)

We turn now to $W_{s,t}$. This problem involves both largest and smallest elements, so for a good bound we must modify the definition of chaos. Pohl's proof that $W_{1,1}(n) = \lceil \frac{3}{2}n \rceil - 2$ [Po] motivates the definition.

An element e is minimal if there is no f, $e > f$. So the maximal and minimal elements e divide into three classes: isolated (no comparison involves e, i.e., e is both maximal and minimal); proper maximal (e is maximal but not minimal) and proper minimal (e is minimal but not maximal). Define functions that count the number of elements in each class: is (P), max (P), and min (P) are the number of isolated, proper maximal, and proper minimal elements in P, respectively.

Define the set $A_k^j(P)$ to contain all consistent assignments defined on $j + k$ elements of $\{e_1,\ldots,e_n\}$ with values in $\{1,\ldots,j\} \cup \{n-k+1,\ldots,n\}$.

Finally to define chaos, set

$$mm\ (P) = \max\ (P) + \min\ (P) + \frac{3}{2}\cdot \text{is}\ (P) - 2,$$

$$C_{s,t}(P) = \sum_{a\in A_{t-1}^{s-1}} 2^{mm(a\cdot P)} \quad .$$

For example, if the partial order P_o contains no relations,

$$C_{s,t}(P_o) = n(n-1)\ldots(n-s-t+3)2^{\frac{3}{2}\cdot(n-s-t)+1} \quad . \tag{6}$$

If in the partial order P_f the s smallest and t largest elements are known, then

$$C_{s,t}(P_f) = 1 \quad . \tag{7}$$

Now we show this function has the non-decreasing property.

Lemma 3: $\quad C_{s,t}(P) \leq C_{s,t}(P_>) + C_{s,t}(P_<)$.

Proof: As in Lemma 2, it suffices to show that for any $a \in A_{t-1}^{s-1}(P)$,

$$2^{mm(a\cdot P)} \leq 2^{mm(a\cdot P_>)} + 2^{mm(a\cdot P_<)} , \tag{8}$$

where a is consistent with both partial orders $P_>$, $P_<$. Without loss of generality, suppose

$$mm(a\cdot P_<) \leq mm(a\cdot P_>) . \tag{9}$$

Now we investigate how mm can decrease going from $a\cdot P$ to $a\cdot P_<$.

The relation $e < f$ can change e's class, thereby decreasing mm. We classify the possible changes below. For each change, we give e's original class in $a\cdot P$, followed in parentheses by the decrease in mm due to the change in e's class.

(i) isolated $(\frac{1}{2})$,

(ii) proper maximal (1),

(iii) no change.

For example in (i), e is isolated in $a\cdot P$; in $a\cdot P_<$, e is proper minimal; so $\underline{is}$ decreases by 1 and $\underline{min}$ increases by 1, giving a net decrease of $\frac{1}{2}$ in $\underline{mm}$.

Similarly a change in f's class, due to the relation $e < f$, decreases mm as follows:

(iv) isolated $(\frac{1}{2})$,

(v) proper minimal (1) ,

(vi) no change.

Note any of (i)-(iii) can occur together with any of (iv)-(vi). We show below that in each case, (8) holds.

First suppose the changes in class together decrease $\underline{mm}$ by 2. Thus (ii) and (v) hold. Clearly the relation $e > f$ does not change $\underline{mm}$, i.e., $mm(a\cdot P) = mm(a\cdot P_>)$, and (8) follows.

Next suppose the changes decrease mm by $\frac{3}{2}$. Thus (i) and (v), or (ii) and (iv), hold. By symmetry, suppose the former. When $e > f$ is added to P, e changes from isolated to proper maximal, while f does not change; thus mm decreases by $\frac{1}{2}$. So the right-hand side of (8) is $(2^{-\frac{3}{2}} + 2^{-\frac{1}{2}})2^{mm(a\cdot P)}$, larger than $1.06\cdot 2^{mm(a\cdot P)}$. Thus (8) holds.

In all other cases, the changes in class together decrease mm by 1 or less. By (9), the right-hand side of (7) is at least $2\cdot 2^{mm(a\cdot P)-1}$. Thus again (8) holds. □

Now we define a demon by the (*) rule. Since Lemma 3 is the analog of Lemma 2, the analog of (4) holds for $C_{s,t}$. Thus we have the following bounds.

Theorem 3: For any s, t, n, where $s,t \geq 1$, and $s + t \leq n$,

$$W_{s,t}(n) \geq \lceil \tfrac{3}{2}(n-s-t) + 1 + \lg n(n-1)\ldots(n-s-t+3) \rceil .$$

If P is a partial order on n elements, then

$$W_{s,t}(P) \geq \lceil \lg C_{s,t}(P) \rceil .$$

Proof: Substitute (6) and (7) in the analog of (4). □

Aside from Pohl's result, the only other bound on $W_{s,t}(n)$ is Floyd's upper bound [Kn, p.220, ex.17],

$$W_{s,t}(n) \leq \lceil \tfrac{3}{2}n \rceil - s - t + \sum_{i=n-s+2}^{n} \lceil \lg i \rceil + \sum_{i=n-t+2}^{n} \lceil \lg i \rceil .$$

This can be slightly improved, as follows.

Lemma 4: For any s,t,n, where $s,t \geq 1$, $s + t \leq n$,

$$W_{s,t}(n) \leq \lceil \tfrac{3}{2}n \rceil - s - t + \sum_{\substack{i=n-s-t+2 \\ i \neq n-\min(s,t)+1}}^{n} \lceil \lg i \rceil \ .$$

Proof: An algorithm for this upper bound is similar to Floyd's. It uses repeated tree selection, the method used for $W_t(n)$ [Kn, p.212]. Without loss of generality, assume $s \leq t$. First find the s smallest elements by tree selection. This uses $n - s + \sum_{i=n-s+2}^{n} \lceil \lg i \rceil$ comparisons. Then find the t largest elements from among the remaining $n - s$ elements. This uses $n - s - t + \sum_{i=n-s-t+2}^{n-s} \lceil \lg i \rceil$ comparisons. However at least $\lfloor \frac{n}{2} \rfloor - s$ of these comparisons are made in the first tree selection and need not be repeated. The total number of comparisons is thus given by the Lemma. □

Now we estimate the accuracy of Theorem 3. Let the lower bound on $W_{s,t}(n)$ be $w_{s,t}(n) = \lceil \frac{3}{2}(n-s-t) + 1 + \lg n(n-1)\ldots(n-s-t+3) \rceil$. Note $w_{1,1}(n)$ is identical to Pohl's bound and so is exact. If $s + t = n$, $w_{s,t}(n)$ is the information-theoretic bound on sorting. In general we have the following.

Corollary 3: For any s,t,n, where $s,t \geq 1$, $s + t \leq n$,

$$w_{s,t}(n) \leq W_{s,t}(n) \leq w_{s,t}(n) + \lfloor \tfrac{3}{2}(s+t) \rfloor - 3.$$

Proof: First note $\frac{3}{2}(n-s-t) + 1 \geq \lceil \frac{3}{2}n \rceil - \lfloor \frac{3}{2}(s+t) \rfloor$. Thus

$$(10) \quad w_{s,t}(n) \geq \lceil \tfrac{3}{2}n \rceil - s - t + \lceil \lg n(n-1)\ldots(n-s-t+3) \rceil - \lfloor \tfrac{1}{2}(s+t) \rfloor \ .$$

Further,

$$(11)\quad \lceil \lg n(n-1)\ldots(n-s-t+3)\rceil + s + t - 3 \geq \sum_{i=n-s-t+3}^{n} \lceil \lg i \rceil \geq \sum_{\substack{i=n-s-t+2 \\ i\neq n-\min(s,t)+1}}^{n} \lceil \lg i \rceil .$$

Combining (10), (11) and Lemma 4 gives the desired upper bound on $W_{s,t}(n)$. □

The chaos function $C_{s,t}$ gives another bound on the General Partition Problem $P(i_1,\ldots,i_k)$, for $k \geq 3$. Choose j in $1 < j < k$, and set $s = i_1 + \ldots + i_{j-1}$, $t = i_{j+1} + \ldots + i_n$; then (4) gives

$$(12)\quad P(i_1,\ldots,i_k) \geq \left\lceil \frac{3}{2} i_j + 1 + \lg\binom{n}{i_1,\ldots,i_k}/(i_{j+1})(i_{j+2}) \right\rceil .$$

The best bound is achieved when $i_j = \max(i_2,\ldots,i_{k-1})$. If $i_j > \frac{2}{3}\max(i_1,i_k)$, then the above bound is usually better than (5). For example in the interesting special case $i_1 = \ldots = i_k = \frac{n}{k}$ (and $k \geq 3$), the lower bound is $n(\lg k + \frac{3}{2k}) + O(\lg n)$.

3. Comparisons with Linear Functions

Much work has been done on sorting and selection problems when the input is numerical, and comparisons involve functions (in particular, linear functions) of the input [DL,Fr,Ra,Re,Sp,Yao]. Lower bounds in this more realistic model of computation can be used to give plausible arguments for the optimality of algorithms in seemingly unrelated areas [e.g., SH]. This section extends the bounds of Section 2 to allow comparisons involving linear functions. The problems considered are $W_t^*(n)$, $V_t^*(n)$, and $U_t^*(n)$.

Suppose an algorithm takes as input a vector of real numbers $\underset{\sim}{e} = (e_1,\ldots,e_n)$, and solves a selection problem by making comparisons involving linear functions of $\underset{\sim}{e}$. We represent the algorithm by a labelled ternary tree, called a linear function tree. Each interior node is labelled with a linear function, $\ell(\underset{\sim}{e}) = \sum_{j=1}^{n} r_j e_j + s$ (r_j and s are real numbers.) The subtrees of the node correspond to the possibilities $\ell(\underset{\sim}{e}) > 0$, $\ell(\underset{\sim}{e}) = 0$, and $\ell(\underset{\sim}{e}) < 0$. Each leaf is labelled with the answer to the selection problem.

A node in a linear function tree is feasible if some input vector leads to it. The following result shows it suffices to consider only feasible nodes.

Lemma 5: For any linear function tree T, there is a linear function tree F for the same problem, with height at most that of T, and all nodes feasible.

Proof: We first show that in T, a feasible interior node has either one or three feasible sons. To do this, let x be a feasible node with two feasible sons. We show the third son is feasible.

Let $\underset{\sim}{f}$ and $\underset{\sim}{g}$ be input vectors leading to two feasible sons of x. It is easy to see for some $\epsilon > 0$ and every θ in $-\epsilon < \theta < 1 + \epsilon$, the vector $\underset{\sim}{h} = \theta\underset{\sim}{f} + (1-\theta)\underset{\sim}{g}$ leads to a son of x. Let ℓ be the label of x. If $\ell(\underset{\sim}{f}) > 0$ and $\ell(\underset{\sim}{g}) = 0$, then $\ell(\underset{\sim}{h}) < 0$ for $\theta < 0$; if $\ell(\underset{\sim}{f}) > 0$ and $\ell(\underset{\sim}{g}) < 0$, then $\ell(\underset{\sim}{h}) = 0$ for some θ in $0 < \theta < 1$. These two cases imply that in general, all three sons of x are feasible.

To construct the tree F, prune T as follows: Repeatedly remove a node with only one feasible son, and replace it by that son. It is easy to see F has the desired properties. □

In dealing with linear function trees, it is useful to consider the reduced tree ([Yao]). This tree is formed by removing all subtrees for outcomes $\ell(\underset{\sim}{e}) = 0$; so an interior node has two sons, corresponding to $\ell(\underset{\sim}{e}) < 0$ and $\ell(\underset{\sim}{e}) > 0$.

Now we prove the analog of Lemma 1 for comparisons involving linear functions.

Lemma 6: A linear function tree for $W_1^*(n)$ has at least 3^{n-1} leaves, and its reduced tree has at least 2^{n-1} leaves.

Proof: Reingold [Re] shows by linear algebra that if a leaf is feasible, its distance from the root is at least $n - 1$. Lemma 5 shows we can assume all leaves are feasible. The Lemma follows easily. □

The bound on $W_t^*(n)$ is analogous to Theorem 1.

Theorem 4: For any t, n, where $1 \le t \le n$,

$$W_t^*(n) \ge n - t + \lceil \lg n(n-1)\ldots(n-t+2) \rceil .$$

Proof: Let T be a linear function tree for $W_t^*(n)$, and let R be its reduced tree. It suffices to show R contains $n(n-1)\ldots(n-t+2)2^{n-t}$ leaves, since R is binary and is included in T. To do this, we show (as in Theorem 1) every sequence of $t-1$ indices is listed as the $t-1$ largest indices in at least 2^{n-t} leaves. Since there are $n(n-1)\cdot\ldots\cdot(n-t+2)$ sequences, the Theorem follows.

For ease of notation, let the sequence of $t - 1$ largest indices be $n,n-1,\ldots,n-t+2$. Prune R to S, a reduced tree for an input vector $(e_1,\ldots,e_{n-t+1})$, as follows. Suppose an interior node x has label $\ell(\underset{\sim}{e}) = \sum_{j=1}^{n} r_j e_j + s$, where some coefficient r_j with $j \geq n - t + 2$ is non-zero. Let k be the largest index for such a coefficient. If $r_k > 0$, replace x by the son corresponding to $\ell(\underset{\sim}{e}) > 0$. Similarly if $r_k < 0$, use the son for $\ell(\underset{\sim}{e}) < 0$. Repeat this for each such node x. The resulting tree is S.

Consider an arbitrary vector $\underset{\sim}{e} = (e_1,\ldots,e_{n-t+1})$. If M is sufficiently large, the vector $\underset{\sim}{f} = (e_1,\ldots,e_{n-t+1},M,M^2,\ldots,M^{t-1})$ leads to the same leaf in R as $\underset{\sim}{e}$ leads to in S. (M is easily computed as a function of $\max\{|e_i|,|r_j|,|s|\}$, and $\min\{|r_j| \mid r_j \neq 0\}$.)

The label of the leaf in R lists the t largest elements of $\underset{\sim}{f}$. Thus a leaf in S lists $n,\ldots,n-t+2$ as the $t - 1$ largest indices. The t^{th} largest index of $\underset{\sim}{f}$ is the largest index of $\underset{\sim}{e}$. Thus S is essentially a reduced tree that finds the largest of $n - t + 1$ elements. Lemma 6 shows S contains 2^{n-t} leaves, as desired. □

The bound on $U_t^*(n)$ is analogous to Corollary 2.

Theorem 5: For any t,n, where $1 \leq t \leq n$,

$$U_t^*(n) \geq n - t + \lceil \lg(\binom{n}{t-1}/t) \rceil .$$

Proof: Let R be a reduced tree for $U_t^*(n)$. It suffices to show each set of $t - 1$ indices is included in the set of t largest indices in at least 2^{n-t} leaves. For there are $\binom{n}{t-1}$ sets of

$t - 1$ indices. A given leaf is counted for at most t of these sets. So there are at least $\binom{n}{t-1}2^{n-t}/t$ leaves. This implies Theorem 5.

For ease of notation let the set of $t - 1$ indices be $\{n,n-1,\ldots,n-t+2\}$. We proceed as in Theorem 4, pruning R to a reduced tree S for $W_1^*(n-t+1)$. (The sequence $(e_1,\ldots,e_{n-t+1})$ corresponds to $(e_1,\ldots,e_{n-t+1},M,M^2,\ldots,M^{t-1})$.) Then S contains 2^{n-t} leaves, as desired. □

The bound on $V_t^*(n)$ is also analogous to Corollary 2. The argument below shows $V_t^*(n) \geq U_t^*(n)$, a fact first observed by Yao [Yao].

Theorem 6: For any t,n, where $1 \leq t \leq n$,

$$V_t^*(n) \geq n - t + \lceil \lg \binom{n}{t-1} \rceil .$$

Proof: Let R be a reduced tree for $V_t^*(n)$. A leaf of R specifies the t^{th} largest index. We show that in addition, the set of $t - 1$ largest indices is known. (Note from Figure 4 this is true for $V_t(n)$.) This suffices to prove Theorem 6, since it implies there are 2^{n-t} leaves for each of the possible $\binom{n}{t-1}$ sets of $t - 1$ largest indices. (The argument is analogous to Theorem 5.)

To show the $t - 1$ largest indices are known, we argue by contradiction. Let y be a leaf of R. For ease of notation, let 1 be the label of y, i.e., e_1 is the t^{th} largest element in a vector leading to y. Let $\underset{\sim}{f}$ and $\underset{\sim}{g}$ be vectors leading to y, whose sets of $t - 1$ largest indices differ. Again for ease of notation, assume $f_1 < f_2$, $g_1 > g_2$.

For every θ in $0 \le \theta \le 1$, the vector $\underset{\sim}{h} = \theta\underset{\sim}{f} + (1-\theta)\underset{\sim}{g}$ leads to y. Let Ψ be the unique value of θ in $0 < \theta < 1$ where the first two components of $\underset{\sim}{h}$ agree, $h_1 = h_2$. For $\theta < \Psi$, $h_1 > h_2$, so h_2 is not one of the $t - 1$ largest elements. It is easy to see that for $\theta = \Psi$, there are $t - 1$ elements h_i, $i \neq 1,2$, with value $h_i \ge h_1 = h_2$. These $t - 1$ elements h_i, and also h_2, can be increased slightly so each is greater than h_1, and the resulting vector $\underset{\sim}{h}'$ still leads to y. (Note all comparisons on the path leading to y have outcomes $>$ or $<$.) But h_1' is not the t^{th} largest element of $\underset{\sim}{h}'$. Thus tree R gives a wrong answer. This is the desired contradiction. □

Note if the tree is not reduced, the $t - 1$ largest indices are <u>not</u> known at a leaf. For example, the median of (e_1,e_2,e_3) can be found in one comparison if $2e_1 - e_2 - e_3 = 0$. However it is not known whether e_2 or e_3 is the largest element.

The only other bound on any of these linear selection functions is due to Yao [Yao],

$$V_t^*(n) \ge n + t - 2, \quad \text{for } t \le n/2.$$

Theorem 6 is superior for small t; Yao's bound is better at the median, by a lower order term $O(\lg n)$.

Theorems 4-6 can be extended to get bounds analogous to Theorem 2 for $W_t^*(P)$, $U_t^*(P)$, and $V_t^*(P)$. A more interesting selection problem is when, instead of P, we are given a collection of relations, $\ell(\underset{\sim}{e}) > 0$. It may be possible to define a chaos function for this problem, thus obtaining a lower bound. However we have not been able to carry out this approach. Also open is the problem of finding a good bound on $W_{s,t}^*$.

References

[BFPRT] Blum, M., R. Floyd, V. Pratt, R. Rivest, and R. Tarjan, "Time bounds for selection", J. Comput. Systems Sci. 7 (1973), pp. 448-461.

[DL] Dobkin, D. and R. Lipton, "On the complexity of computations under varying sets of primitives", Research Rept. 42, Dept. of Comp. Sci., Yale University, 1975.

[DM] Dobkin, D. and J. I. Munro, "Time and space bounds for selection problems", preprint.

[FG] Fussenegger, F., H. N. Gabow, "Using comparison trees to derive lower bounds for selection problems", Proc. 17th Annual Symp. on Foundations of Comp. Sci., Houston, Texas, 1976, pp. 178-182.

[Fr] Friedman, N., "Some results on the effect of arithmetics on comparison problems", Proc. 13th Annual Symp. on Switching and Automata Theory, College Park, Maryland, 1972, pp.139-143.

[H] Hyafil, L., "Bounds for selection", SIAM J. Comput. 5 (1976), pp. 109-114.

[Kir] Kirkpatrick, D. G., "Topics in the complexity of combinatorial algorithms", Technical Report 74, University of Toronto, 1974.

[Kis] Kislitsyn, S. S., "On the selection of the k-th element of an ordered set by pairwise comparisons", Sibirskiĭ Mat. Zhurnal 5 (1964), pp. 557-564 (Russian).

[Kn] Knuth, D. E., The Art of Computer Programming, Vol. 3: Sorting and Searching, Addison-Wesley, Reading, Mass., 1973, pp. 209-220.

[M] Malanowicz, K. R., "Improving lower bounds for selection problems", M.S. Thesis, Dept. of Comp. Sci., University of Colorado, 1977.

[Po] Pohl, I., "A sorting problem and its complexity", CACM 15 (1972), pp. 462-464.

[PY] Pratt, V. and Yao, F., "On lower bounds for computing the i-th largest element", Proc. 14th Annual Symp. on Switching and Automata Theory, Iowa City, Iowa, 1973, pp. 70-81.

[Ra] Rabin, M. O., "Proving simultaneous positivity of linear forms", J. Comput. Systems Sci. 6 (1972), pp. 639-650.

[Re] Reingold, E.,"On the optimality of some set algorithms", J. ACM 19 (1972), pp. 649-659.

[SH] Shamos, M. I. and D. Hoey, "Closest-point problems", Proc. 16th Annual Symp. on Foundations of Comp. Sci., Berkeley, Calif. 1975, pp. 151-162.

[Sp] Spira, P. M., "Complete linear proofs of systems of linear inequalities", J. Comput. Systems Sci. 6 (1972), pp. 205-216.

[SPP] Schönhage, A., Paterson, M., and Pippenger, N., "Finding the median", J. Comput. Systems Sci 13 (1976), pp. 184-199.

[W] Wells, M. B., "Applications of a language for computing incombinatorics", Proc. IFIP Congress 65 2, New York, N.Y., 1965, pp. 497-498.

[Yao] Yao, A., "On the complexity of comparison problems using linear functions", Proc. 16th Annual Symp. on Foundations of Comp. Sci., Berkeley, Calif., 1975, pp. 85-89.

[Yap] Yap, C. K., "New lower bounds for median and related problems", Research Report 79, Yale University, 1976.

Figures

Figure 1: A comparison tree for $W_2(4)$.

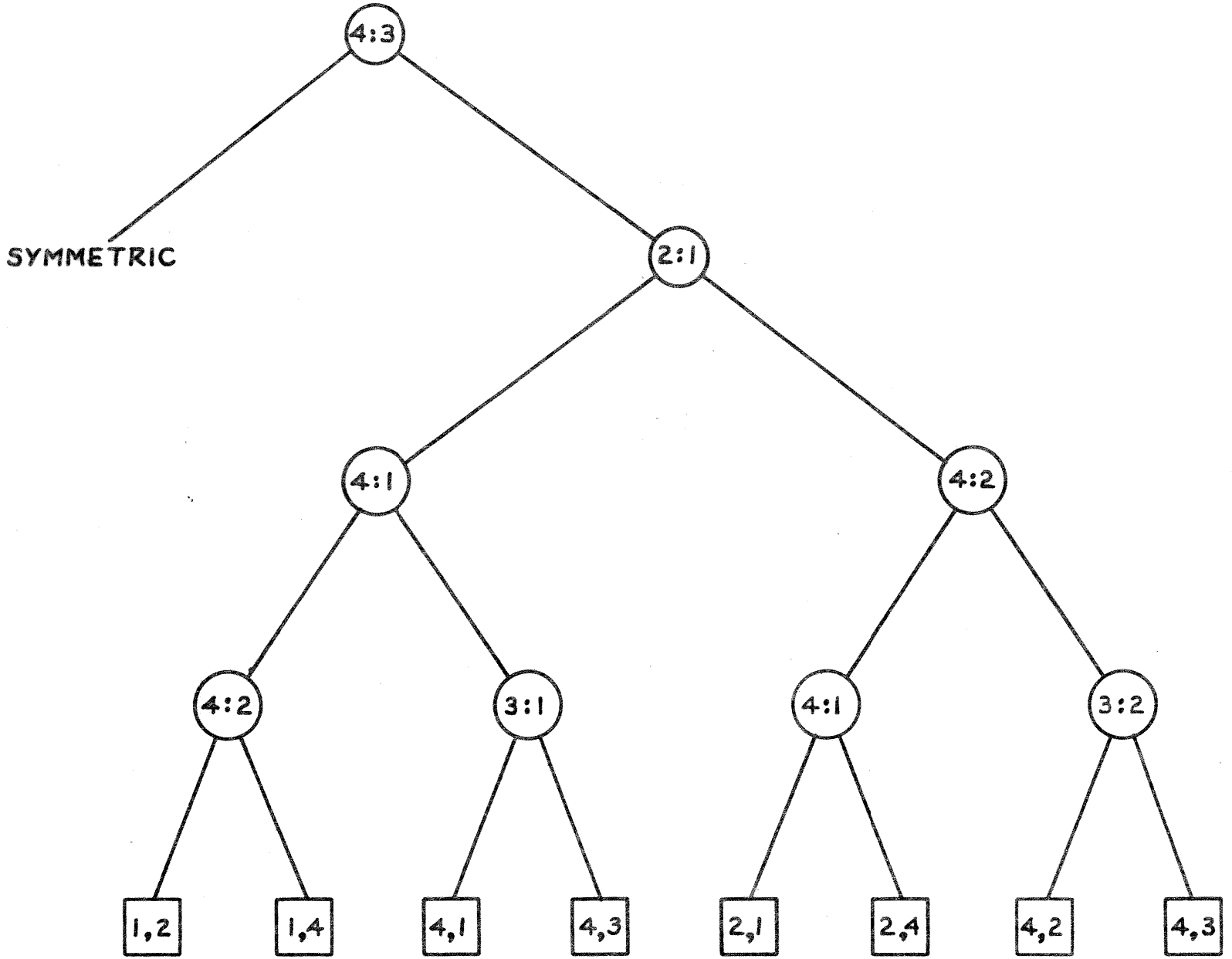
Figure 2: The tree of Figure 1 pruned so 4 is always the maximum element.

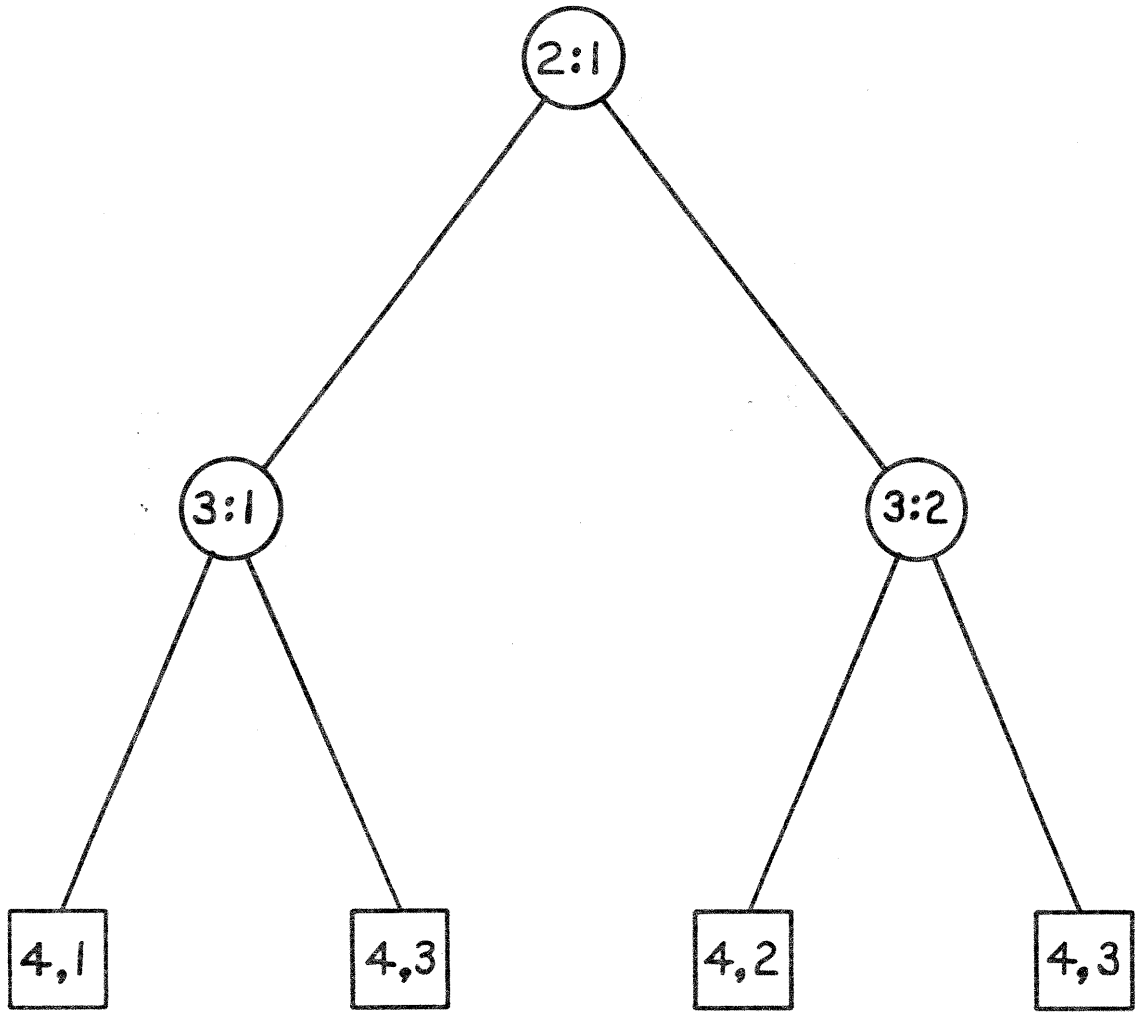
Figure 3: The Hasse diagram for the leftmost interior node in Figure 1.

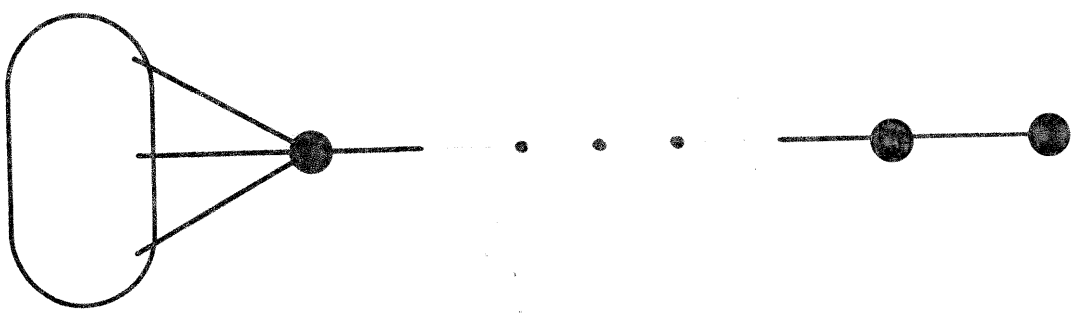
Figure 4: P_f for the W_t problem: The vertical chain has t elements.

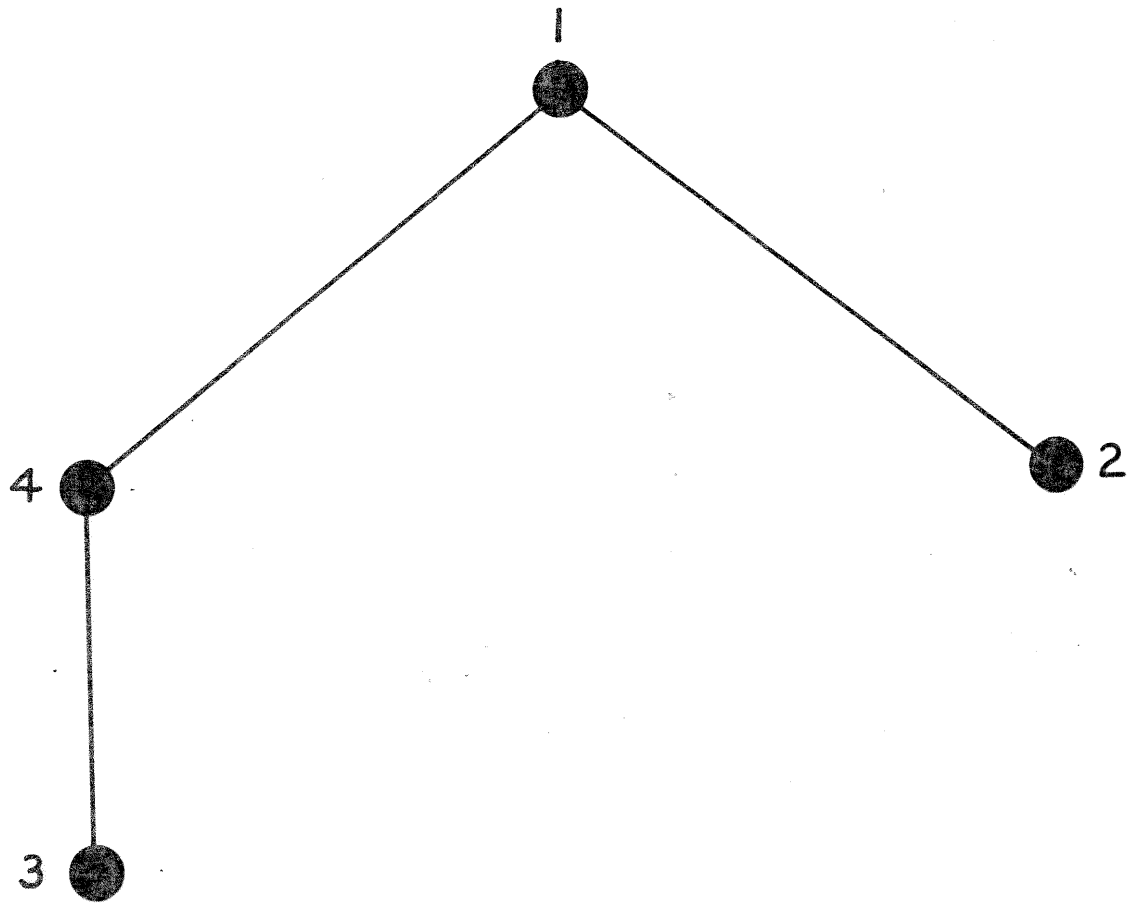
Figure 5: (a) P_f for V_t: The top set has $t - 1$ elements, the bottom $n - t$.

(b) P_f for U_t: The top set has t elements, the bottom $n - t$.


4:3
SYMMETRIC
2:1
4:1
4:2
4:2
3:1
4:1
3:2
1,2
1,4
4,1
4,3
2,1
2,4
4,2
4,3


2:1
3:1
3:2
4,1
4,3
4,2
4,3




1
2
3
4