Astérisque

A. SCHÖNHAGE

The production of partial orders

Astérisque, tome 38-39 (1976), p. 229-246

<http://www.numdam.org/item?id=AST_1976__38-39__229_0>

Société Mathématique de France
Astérisque 38-39 (1976) p.229-246

THE PRODUCTION OF PARTIAL ORDERS (1)

by

A. SCHÖNHAGE

ABSTRACT

Many of the well-known sorting problems can be understood as the task of producing certain partial orders. We investigate how the cost of such a task depends on the size of the reservoir of elements and upon the number of copies of the partial order to be produced. A reduction technique enables us to obtain lower bounds for several problems of this kind.

1.- INTRODUCTION

In this paper we always assume that we are given a totally ordered finite set R, the reservoir. The order is not known initially and can only be determined by performing successive pair-wise comparisons between elements of R. By branching on the outcome of such comparisons the algorithms under consideration will have binary tree structure. We will only discuss the cost function given by the maximal path length, i.e. the number of comparisons required in the worst case.

In order to motivate the formal concepts of this paper let us first consider some of the well-known sorting problems.

------

(1) This paper was partially supported by The National Science Foundation under research grant GJ-34671 to MIT Project MAC.

Sorting of n elements can be understood as the task of producing a totally ordered string of length n, starting from n singletons. We denote this "partial" order by T_n. Accordingly, each singleton is a T_1, and in this case, production of T_n means the transition from $n \cdot T_1$ to T_n. A simple information theoretical argument yields the lower bound :

$$\sigma(n) \geqslant \lceil \log_2(n!) \rceil \tag{1.1}$$

for the cost $\sigma(n)$ of any optimal algorithm. The best known upper bound :

$$\sigma(n) \leqslant \sum_{k=2}^{n} \lceil \log(\tfrac{3}{4}k) \rceil \tag{1.2}$$

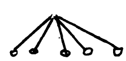
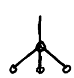
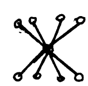
comes from the Ford & Johnson algorithm [1] (see also [3], section 5.3.1). Thus there is still a gap of order n between the two bounds.

Merging of m and n elements means the transition from $T_m + T_n$ to T_{m+n}. In some cases the corresponding cost function $\mu(m,n)$ is known explicitly :

$$\begin{aligned} &\mu(m,n) = m + n - 1 \quad \text{for} \quad |m - n| \leqslant 1, \\ &\mu(1,n) = \lceil \log_2(n+1) \rceil \\ &\mu(2,n) = \lceil \log_2(\tfrac{7}{12}(n+1)) \rceil + \lceil \log_2(\tfrac{14}{17}(n+1)) \rceil. \end{aligned} \tag{1.3}$$

The latter formula (cf. [2]) gives some idea how intricate the answer to fairly simple problems of this type can be. Here the merging problem serves as an example, where the algorithms start from some prescribed partial order.

Selecting the i-th of n elements can be viewed as the production of S_{n-i}^{i-1} from n singletons, where S_m^k denotes a partial order on $m + 1 + k$ elements with one particular element, the centre, which is less than each of k other elements and greater than each of the m remaining elements.

Examples are : S_5^o =  S_3^1 =  S_4^4 = 

Production of S^{o}_{n-1} means to determine the maximum of n elements. Another particular case is the determination of the median of $n = 2k+1$ elements by producing S^{k}_{k}.

We denote the cost function for the "i-th of n problem" by $V_i(n)$ and mention the following results (see [4],[5], and [3], section 5.3.3 for further references and comments) :

$$V_1(n) = V_n(n) = n - 1 \tag{1.4}$$

$$V_2(n) = n - 2 + \lceil \log_2 n \rceil \tag{1.5}$$

and for the median $(n = 2k+1)$

$$1.75\,n - 2 - \log_2 n \leqslant V_{k+1}(n) \leqslant 3\,n + o(n). \tag{1.6}$$

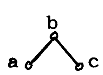
Our present work was mainly stimulated by questions arising from the median problem. In particular, the concept of mass production as treated in section 5 proved to be extremely useful for obtaining the upper bound $\sim 3\,n$. The aim of the following sections is to provide a theoretical framework for the new concepts, which will be exemplified by several examples and theorems.

2.- THE NORMAL FORM OF PRODUCTION PROBLEMS

For the formal description of what we unsterstand by the production of a partial order P our basic notion is that of an order preserving embedding (i.e. a 1-1-map) :

$$E : P \underset{op}{\longrightarrow} R$$

into the reservoir R. It should be kept in mind that comparisons can be performed only between the elements of R, whereas the set P with its partial order merely serves as a pattern. When discussing examples we will frequently use a description of such E's as in the following example :



means an E from $P = S^{o}_{2}$ into $R = \{a, b, c, \ldots\}$, where $a < b, c < b$.

A single comparison x?y produces a pair, more precisely one of the embeddings

$\begin{matrix} x \\ | \\ y \end{matrix}$ or $\begin{matrix} y \\ | \\ x \end{matrix}$. In production algorithms the comparisons are used to extend such embeddings step by step. Given some partial order A and an $E : A \xrightarrow[\text{op}]{} R$, the next comparison x?y yields either $E' : A' \xrightarrow[\text{op}]{} R$ or $E'' : A'' \xrightarrow[\text{op}]{} R$, corresponding to the possible cases $x < y$, $x > y$. If $x,y \in E(A)$, then we have $E' = E'' = E$. Otherwise we have to introduce one or two new elements ξ, η, and $E' = E''$ is the extension of E to a mapping from $\tilde{A} = A \cup \{\xi, \eta\}$ into R such that $E'(\xi) = x$, $E'(\eta) = y$ in any case. Then A' (or A'') is the smallest partial order on the underlying set $\tilde{A}$ that contains the partial order A and $\xi < \eta$ (or $\xi > \eta$, respectively).

Example : Given $\begin{matrix} x \\ | \\ z \end{matrix}$ the comparison x?y yields either $\begin{matrix} y \\ | \\ x \\ | \\ z \end{matrix}$ or $\begin{matrix} x & \\ | & \diagdown \\ z & \quad y \end{matrix}$, i.e. $A' = S_1^1 = T_3$, and $A'' = S_2^o$. We can understand this comparison as a final step in the production of an S_2^o, because A' contains $\begin{matrix} & y & \\ z \diagup & & \diagdown x \end{matrix}$. Thus we are led to the following definition : A production algorithm π is a finite binary tree with a partial order B as its unique root and branchings $A \to A', A''$ as explained before. π is said to produce P from B, if for every end-point A of π there is an order preserving embedding $P \xrightarrow[\text{op}]{} A \xrightarrow[\text{op}]{} R$.

If the reservoir R contains more elements than B, then we can extend B by $q = |R| - |B|$ many extra singletons, and $B' = B + q \cdot T_1$ instead of B will cause no essential difference. Therefore, we can always assume $|B| = |R|$. Another important step is to replace P by the set $\langle P \rangle$ of all partial orders A of size $|A| = |B|$ with $P \xrightarrow[\text{op}]{} A$. More generally, we consider nonempty sets $\mathcal{U}$ of partial orders A with $|A| = |B|$ that are closed under extension, i.e. $A \in \mathcal{U}$, $A \xrightarrow[\text{op}]{} A'$ and $|A'| = |A|$ implies $A' \in \mathcal{U}$. A production algorithm π is said to produce $\mathcal{U}$ from B, if π has the root B and if all end-points A of π belong to $\mathcal{U}$.

With respect to this normal form of production problems we define the cost functions :

$$\ell(\pi) := \text{maximal path length of } \pi$$

$$\lambda(\mathcal{U}|B) := \min\{\ell(\pi)\,|\,\pi \text{ produces } \mathcal{U} \text{ from } B\}, \tag{2.1}$$

$$\lambda(P|B) := \lambda(\langle P\rangle|B). \tag{2.2}$$

3.- THE INFLUENCE OF THE SIZE OF THE RESERVOIR

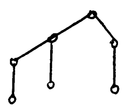
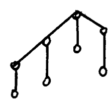
For the common case $B = r \cdot T_1$ with r elements in the reservoir and $|P| = p \leqslant r = p + m$ we use the notation :

$$\lambda_m(P) = \lambda(P|(p+m)\cdot T_1) \qquad (0 \leqslant m). \tag{3.1}$$

Clearly, we have :

$$\lambda_o(P) \geqslant \lambda_1(P) \geqslant \lambda_2(P) \geqslant \ldots = \ldots =: \lambda_\infty(P), \tag{3.2}$$

and it seems to be rather convincing that extra elements cannot facilitate the production of P, i.e. $\lambda_o(P) = \lambda_\infty(P)$. This, however, is not true in general ! M. Paterson has found a rather simple counter-example : for

P =  P' = 

we obtain $\lambda_o(P) = 8$, but $\lambda_1(P) = \lambda_o(P') = 7$.

F. Yao discussed the hypothesis:

$$\lambda_o(S^k_m) = \lambda_\infty(S^k_m) \qquad \text{for all } k,m. \tag{3.4}$$

Here no counter-example is known. The importance of such a plain condition can be judged from the fact that it implies :

$$\lambda_o(S^k_k) \leqslant 5\,k \qquad \text{for all } k, \tag{3.5}$$

and this estimate would imply that the median of n elements could be determined by less than 2 .5n comparisons.

The proof of (3.5) under the hypothesis (3.4) is based upon inequalities like :

$$\lambda_k(S_k^{2k+1}) \leqslant 2k+1+\lambda_o(S_k^k)\ ,$$
$$\lambda_{2k+1}(S_{2k+1}^{2k+1}) \leqslant 3k+2+\lambda_o(S_k^{2k+1}), \tag{3.6}$$

that are true in any case (for details see [5]). The crucial point is, whether we can write λ_o also on the left-hand side.

For $k=0$, or $k=1$ Yao's hypothesis is true, as we can show by evaluating $\lambda_\infty(S_m^o)$, $\lambda_\infty(S_m^1)$ explicitly. From (1.4) we have $\lambda_o(S_m^o) = V_1(m+1) = m$.

The lower bound $\lambda_\infty(S_m^o) \geqslant m$, or $\lambda_t(S_m^o) \geqslant m$ for all t, comes from a simple connectivity argument : initially, there are $r = m+1+t$ many singletons. Any partial order A that contains S_m^o consists of at most $|A| - m = t+1$ components. Since each comparison reduces the number of components by at most 1 we are done.

For $k=1$ we give the following more general

<u>THEOREM 3.1</u>.- Let $B = S_{n_1}^o + S_{n_2}^o + \ldots + S_{n_p}^o$ with $n_i \geqslant 0$, $\sum_{i=1}^{p}(n_i+1) = |B| = m+2+t = r$. Then we have :

$$\lambda(S_m^1|B) \geqslant m - \sum_{i=1}^{p} n_i + \lceil \log_2(\sum_{i=1}^{p} 2^{n_i} - t)\rceil, \tag{3.7}$$

and in particular (all $n_i = 0$) :

$$\lambda_t(S_m^1) \geqslant m + \lceil \log_2(m+2)\rceil. \tag{3.8}$$

(For $t=0$ cf. [3], p. 219, exercise 6). Now by (1.5) we get :

$$\lambda_o(S_m^1) = V_2(m+2) = m + \lceil \log_2(m+2)\rceil = \lambda_t(S_m^1)$$

for all t.

We postpone the proof of Theorem 3.1 because it employs the reduction technique developed in section 6.

Finally, we pose some (open) problems :

- for arbitrary t, is there a P with $\lambda_t(P) > \lambda_{t+1}(P)$?

- Can $\lambda_o(P) - \lambda_\infty(P)$ become arbitrarily large ?
- Find nontrivial bounds for the function f defined by :

$$f(n) = \max\{t \mid \lambda_t(P) > \lambda_\infty(P) \text{ where } |P| \leqslant n\} .$$

4.- AN INFORMATION THEORETICAL APPROACH

For any partial order A of size $|A| = r = |R|$ we consider the set $\mathcal{E}(A|R)$ of all order preserving embeddings $E : A \xrightarrow[op]{} R$ and their number $e(A) = \# \mathcal{E}(A|R)$.

For $|A_1| = r_1$, $|A_2| = r_2$ we obtain :

$$e(A_1 + A_2) = \frac{(r_1 + r_2)!}{r_1! \cdot r_2!} \cdot e(A_1) \cdot e(A_2). \tag{4.1}$$

When, by a comparison, A is extended to A' or A", then $\mathcal{E}(A|R)$ splits up into the two disjoint subsets $\mathcal{E}(A'|R)$, and $\mathcal{E}(A''|R)$, hence :

$$e(A) = e(A') + e(A''). \tag{4.2}$$

Therefore, every production algorithm π producing $\mathcal{U}$ from B contains a path :

$$B = A_o \to A_1 \to \dots \to A_t \in \mathcal{U}$$

with the property $e(A_j) \geqslant ½\, e(A_{j-1})$ for $j = 1,\dots,t$.

THEOREM 4.1.- $\lambda(\mathcal{U}|B) \geqslant \lceil \log_2(e(B)/\max_{A \in \mathcal{U}} e(A)) \rceil$.

Given the special case of a partial order P with $|P| = n$ and the reservoir size $r = n + m$ we have to consider $\mathcal{U} = \langle P + m T_1 \rangle$, $B = r \cdot T_1$. Since $P + m \cdot T_1 \xrightarrow[op]{} A$ for $A \in \mathcal{U}$ implies $e(P + m \cdot T_1) \geqslant e(A)$, we have :

$$\max\{e(A) \mid A \in \langle P + m T_1 \rangle\} = e(P + m \cdot T_1).$$

From $e(B) = r!$, $e(m T_1) = m!$, and (4.1) we then deduce that :

$$\mathcal{B}(P) \ := \ (n + m)!/e(P + m \cdot T_1) = n!/e(P) \tag{4.3}$$

does not depend on m. Thus Theorem 4.1 has the

Corollary 4.2.- $\lambda_m(P) \geqslant \lceil \log_2 \mathcal{B}(P) \rceil$ for all m.

For $P = T_n$ this implies (1.1), but more precisely :

$$\lambda_\infty(T_n) \geqslant \lceil \log_2 (n!) \rceil .$$

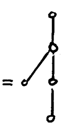
For Paterson's counter-example P in (3.3) we compute $\mathcal{B}(P) = 7 \cdot 2^4$. There is only a small margin compared with 2^7, and $7!$ is an odd multiple of 2^4 only.

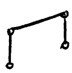
In general, it seems to be promising to analyze the structure of the sets $\mathcal{E}(A|R)$ in more detail.

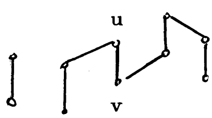
5.- MASS PRODUCTION

Producing many copies of the same partial order P may sometimes allow to save comparisons. The simplest example we know of involves S_3^1.

LEMMA 5.1.- In contrast to $\lambda_\infty(S_3^1) = 6$:

$$\lambda_o(2 \cdot S_3^1) \leqslant \lambda_o(2 \cdot Q) \leqslant 11, \quad \text{where } Q =$$ 

Proof.- Since  can be obtained by 3 comparisons, 8 comparisons are sufficient for :

B = 

The remaining steps $\lambda(2Q|B) \leqslant 3$ are left to the reader (hint : ignore that $v < u$).

We begin the general analysis with some obvious facts. If $|P_2| = p$, then (cf. (3.1)) :

$$\lambda_m(P_1 + P_2) \leqslant \lambda_{m+p}(P_1) + \lambda_m(P_2) \leqslant \lambda_m(P_1) + \lambda_m(P_2), \tag{5.1}$$

and $m \to \infty$ yields :

$$\lambda_\infty(P_1 + P_2) \leqslant \lambda_\infty(P_1) + \lambda_\infty(P_2), \tag{5.2}$$

in particular :

$$\lambda_m(k \,.\, P) \leqslant k \,.\, \lambda_m(P),$$
$$\lambda_\infty(k \,.\, P) \leqslant k \,.\, \lambda_\infty(P). \tag{5.3}$$

The information theoretical quantity $\mathfrak{B}$ behaves similarly, as follows from (4.1) and (4.3) :

$$\mathfrak{B}(P_1 + P_2) = \mathfrak{B}(P_1) \,.\, \mathfrak{B}(P_2) \quad , \quad \mathfrak{B}(k \,.\, P) = \mathfrak{B}(P)^k. \tag{5.4}$$

One can think of several ways to define an asymptotic cost function. Fortunately, the most suggestive versions turn out to have the same values.

<u>THEOREM 5.2</u>. <u>Definition</u>.- For every partial order P the asymptotic cost $\overline{\lambda}(P)$ is defined by :

$$\overline{\lambda}(P) := \inf_k(\lambda_\infty(k \,.\, P)/k) = \inf_k(\lambda_o(k \,.\, P)/k) = \lim_{k\to\infty} (\lambda_o(k \,.\, P)/k).$$

<u>Proof</u>.- Given $\varepsilon > 0$, choose k such that :

$$\lambda_\infty(k \,.\, P)/k \leqslant \overline{\lambda}(P) + \varepsilon \quad , \tag{5.5}$$

and then choose t such that $\lambda_\infty(k \,.\, P) = \lambda_{t.kp}(kP)$, where $p = |P|$. For $m \geqslant tk$, $q = \lceil m/k \rceil$ we have (cf. (5.1)) :

$$\lambda_\infty(m\,P)/m \leqslant \lambda_o(m\,P)/m \leqslant \lambda_o((q+1)k\,P)/(q\,k)$$
$$\leqslant \frac{1}{q} \sum_{j=0}^{q} \lambda_{j.k\,p}(k\,P)/k \leqslant \frac{q+1-t}{q}\, \lambda_{t\,k\,p}(k\,P)/k + \frac{t}{q}\lambda_o(k\,P)/k$$
$$\leqslant \lambda_\infty(k\,P)/k + \frac{t}{q}\lambda_o(P).$$

Now $m \to \infty$, $q \to \infty$ give :

$$\limsup_{m\to\infty} \lambda_\infty(m\,P)/m \leqslant \limsup_{m\to\infty} \lambda_o(m\,P)/m \leqslant \overline{\lambda}(P) + \varepsilon,$$

and $\varepsilon \to 0$ completes the proof.

The information theoretical lower bound also applies to $\overline{\lambda}(P)$. Combining Corollary 4.2 with (5.3), (5.4) we obtain :

<u>Corollary 5.3</u>.- $\overline{\lambda}(P) \geqslant \log_2 \mathcal{B}(P)$

Here we like to mention another nice example (again due to M. Paterson) for savings in mass production : for totally ordered strings of length 7 he showed $\lambda_o(5 \cdot T_7) \leqslant 64$, whereas $\lambda_o(T_7) = \lambda_\infty(T_7) = 13$. This together with Corollary 5.3 gives :

$$12 . 3 \approx \log_2(7!) \leqslant \overline{\lambda}(T_7) \leqslant 12 . 8. \tag{5.6}$$

In connexion with our median algorithm (cf. [5], Theorem 10.1) we obtained the asymptotic estimate :

$$\overline{\lambda}(S_k^k) \leqslant 3 . 5 k + o(k) . \tag{5.7}$$

It coincides remarkably with the lower bound (1.6) for $V_{k+1}(n) = \lambda_o(S_k^k)$. For at least one of these quantities $\lambda_o, \overline{\lambda}$ the 1 . 75 per element should be the true constant !

The reader will notice that, in constrast to $\lambda_o(P)$, or $\lambda_\infty(P)$, the quantity $\overline{\lambda}(P)$ cannot be determined by simply checking finitely many cases. For each single P we have to discuss an infinite sequence of problems. Therefore, even small P's can present considerable difficulties, and so far there are only few examples, where we know the precise value of $\overline{\lambda}(P)$.

<u>THEOREM 5.4</u>.-

$$\overline{\lambda}(S_n^o) = n , \tag{5.8}$$

$$\overline{\lambda}(S_1^1) = 3 \quad , \quad \overline{\lambda}(S_2^1) = 4 \quad , \quad \overline{\lambda}(S_2^2) = 6 , \tag{5.9}$$

$$\overline{\lambda}(S_3^1) = \overline{\lambda}(Q) = 5 . 5 , \tag{5.10}$$

where Q is defined as in Lemma 5.1.

Again the proofs of (5.9), (5.10) will be given later. The proposition (5.8) follows from $\lambda_o(k \cdot S_n^o) = k n$, and this can be shown by the connectivity argument that we already used for the case $k = 1$ in section 3.

Open problems : Is there a partial order P such that $\overline{\lambda}(P)$ becomes an irrational number ? On the contrary, is there always a suitable k such that $\overline{\lambda}(P) = \lambda_\infty(k P)/k$?

- Try to reduce the O(n) gap between (1.1) and (1.2) for $\overline{\lambda}(T_n)$ instead of $\lambda_o(T_n)$.

6.- A REDUCTION TECHNIQUE

Until now we have considered only the special case $\mathcal{U} = < P >$. In this section, however, we have to deal with the general case of a set $\mathcal{U}$ of partial orders (all of the same size $|A| = |B|$) that is closed under extension. Since $\xrightarrow[op]{}$ defines a partial ordering on $\mathcal{U}$, it is sufficient to consider the minimal elements $A_1, A_2, \ldots$ of $\mathcal{U}$, which then generate :

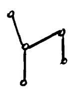
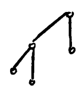
$$\mathcal{U} = \bigcup_j < A_j > = : < A_1, \ldots, A_v > \tag{6.1}$$

$$= \{A \mid |A| = |B| \quad \text{and} \quad A_j \xrightarrow[op]{} A \quad \text{for some } j\}.$$

In view of the definitions (2.1), (2.2) the reader may conjecture that :

$$\lambda(\mathcal{U} \mid B) \leqslant \min_j \lambda(A_j | B) \tag{6.2}$$

is always an equality, but there is a simple counter-example : for

$A_1 =$  $\qquad\qquad A_2 =$  (6.3)

we have $\lambda_o(A_1) = \lambda_o(A_2) = 5$, but $\lambda(< A_1, A_2 > | 5 \cdot T_1) = 4$.

For any partial order A let min A (max A) denote the set of all elements $\alpha \in A$ with $\alpha > \beta$ $(\alpha < \beta)$ for no β in A. If A is finite and nonempty, then also min $A \neq \emptyset$, max $A \neq \emptyset$. We define the two processes called "min-reduction" and "max-reduction" by

$$\begin{aligned} \mathrm{MIR}(A) &: = \{A' = A \setminus \{\alpha\} \mid \alpha \in \min A\}, \\ \mathrm{MAR}(A) &: = \{A' = A \setminus \{\alpha\} \mid \alpha \in \max A\}, \end{aligned} \tag{6.4}$$

$$\begin{aligned} \mathrm{MIR}\,\mathcal{U} &: = \bigcup_{A \in \mathcal{U}} \mathrm{MIR}(A), \\ \mathrm{MAR}\,\mathcal{U} &: = \bigcup_{A \in \mathcal{U}} \mathrm{MAR}(A). \end{aligned} \tag{6.5}$$

We state without proof :

LEMMA 6.1.- If $\mathcal{U} = < A_1,\dots,A_v >$ and $\bigcup_j \mathrm{MIR}(A_j) = \{A'_1,\dots,A'_q\}$, then :

$$\mathrm{MIR}\,\mathcal{U} = < A'_1,\dots,A'_q >.$$

An analogous statement applies to MAR, by virtue of duality. These reductions will be our main tool for the proof of a general theorem that will be used then to prove Theorems 3.1 and 5.4.

In section 2, we have explained the nature of comparisons in production algorithms. More precisely, the assumption that nothing is known about the total order of the reservoir R in advance has the following technical meaning : given $E : B \xrightarrow[\mathrm{op}]{} R$, the choice of the elements x and y for the next comparison can only depend on the structure of B, i.e. two elements $\xi \neq \eta$ are selected from B which are still unordered in B (otherwise the comparison would be redundant), and E then determines $x = E(\xi)$, $y = E(\eta)$ in R. The idea of min-reduction is in case of $\xi \in \min B$ to restrict the further analysis to those embeddings E which map ξ onto the minimum of R. Then it is possible to omit ξ and $E(\xi) = x = \min R$, thus obtaining a smaller problem.

LEMMA 6.2.- If $\xi \in \min B$, $B' = B \setminus (\xi)$, $\mathcal{U}' = \mathrm{MIR}\,\mathcal{U}$, then :

$$\lambda(\mathcal{U} \mid B) \geqslant \lambda(\mathcal{U}' \mid B').$$

Proof.- Let π be an optimal algorithm producing $\mathcal{U}$ from B, where $E : B \xrightarrow[\mathrm{op}]{} R$ is given such that $E(\xi) = \min R$. Then π terminates with some $A \in \mathcal{U}$, and $E : A \to R$ is still order preserving, therefore $\xi \in \min A$, and $A' = A \setminus \{\xi\} \in \mathcal{U}'$. If we delete from π all comparisons that involve $x = \min R$ and all branches belonging to outcomes $x > y$, then we obtain a reduced algorithm π' with root B' that produces $\mathcal{U}'$, hence :

$$\lambda(\mathcal{U} \mid B) = \ell(\pi) \geqslant \ell(\pi') \geqslant \lambda(\mathcal{U}' \mid B').$$

Sometimes we will need multiple min-reduction. Repeated application of Lemma 6.2 gives :

LEMMA 6.3.- Let $B_o \subseteq B$, $B' = B \setminus B_o$, $\mathcal{U}' = MIR^s \mathcal{U}$, where $s = |B_o| \geq 1$. If there is no pair $\xi > \eta$ with $\xi \in B_o$, $\eta \in B'$, then :

$$\lambda(\mathcal{U} \mid B) \geq \lambda(\mathcal{U}' \mid B').$$

Again, similar results hold for max-reduction.

Our main theorem deals with a funtion $f : D \to \mathbb{R}$, where the domain D contains pairs of partial orders (A,B) with $|A| = |B|$. $f(A|B)$ is intended as a measure for the complexity of producing A from B. B will vary in some set $\mathcal{B}$ and A in a set $\mathcal{G}$ of partial orders of variable size.

THEOREM 6.4.- Assume that two sets $\mathcal{B}$ and $\mathcal{G}$ of finite partial orders and a function $f : D \to \mathbb{R}$ with domain $D = \{(A,B) \in \mathcal{G} \times \mathcal{B} \mid |A| = |B|\}$ satisfy the following conditions :

C0 : $A \in \mathcal{G} \Rightarrow MIR(A) \subseteq \mathcal{G}$ and $MAR(A) \subseteq \mathcal{G}$.

C1 : $(A,B) \in D$ and $A \xrightarrow[op]{} B \Rightarrow f(A|B) \leq 0$.

C23 : For every $B \in \mathcal{B}$ and arbitrary $\xi, \eta \in B$ at least one of extensions, say B^*, obtained from B by adding either $\xi < \eta$, or $\xi > \eta$, satisfies C2 or C3 for all $A \in \mathcal{G}$ with $|A| = |B|$ and $\lambda(A|B) \geq 1$:

C2 : $B^* \in \mathcal{B}$ and $f(A|B) \leq f(A|B^*) + 1$;

C3 : $B' \in \mathcal{B}$ and $f(A|B) \leq f(A'|B') + 1$ for all $A' \in MIR^s(A)$ (all $A' \in MAR^s(A)$), where $B' = B^* \setminus B^*_o$ is obtained from B^* by a suitable (multiple) min-reduction (or max-reduction).

Then for every $\mathcal{U} = \langle A_1, \dots, A_v \rangle$ and B with $(A_i, B) \in D$ for all i we have the lower bound :

$$\lambda(\mathcal{U} \mid B) \geq \min_{1 \leq i \leq v} f(A_i|B). \tag{6.6}$$

The proof is by induction on $n = \lambda(\mathcal{U} \mid B)$. For $n = 0$ we use C1. For $n > 0$ let π be an optimal algorithm that produces $\mathcal{U}$ from B. Its first comparison specifies $\xi, \eta \in B$, such that we can choose B^* according to C23. If B^* satisfies C2, we can apply the induction hypothesis (6.6) to $\mathcal{U}$, B^*, because $n = \ell(\pi) \geqslant 1 + \lambda(\mathcal{U} \mid B^*)$, hence $\exists_j \; \lambda(\mathcal{U} \mid B) \geqslant 1 + f(A_j \mid B^*) \geqslant f(A_j \mid B) \geqslant \min_i f(A_i \mid B)$.

Otherwise C3 holds for B^*, and Lemma 6.3 yields

$$n = \lambda(\mathcal{U} \mid B) \geqslant 1 + \lambda(\mathcal{U} \mid B^*) \geqslant 1 + \lambda(\mathcal{U}' \mid B').$$

This time (6.6) can be applied to $\mathcal{U}'$, B'. By virtue of Lemma 6.1 and C3 there is a $j \leqslant v$ and an $A' \in MIR^s(A_j)$ (or $A' \in MAR^s(A_j)$) such that :

$$\lambda(\mathcal{U} \mid B) \geqslant 1 + f(A' \mid B') \geqslant f(A_j \mid B) \geqslant \min_i f(A_i \mid B).$$

7.-PROOF OF THEOREM 3.1

This first application of Theorem 6.4 employs only min-reduction. Therefore we can modify condition CO by omitting MAR. We put :

$$\mathcal{U} := \{A_{m,t} = S^1_m + t \cdot T_1 \mid m,t \geqslant 0\} \cup \{r T_1 \mid r \geqslant 0\},$$

$$\mathfrak{B} := \{B = S^o_{n_1} + \ldots + S^o_{n_p} \mid n_i \geqslant 0\},$$

$$f(A_{m,t} \mid B) := m - \sum_{i=1}^{p} n_i + \lceil \log_2 (\sum_{i=1}^{p} 2^{n_i} - t) \rceil, \tag{7.1}$$

$$f(r \cdot T_1 \mid B) := 0,$$

where :

$$m + 2 + t = \sum_{i=1}^{p} (n_i + 1) = r. \tag{7.2}$$

<u>C1</u> : $A_{m,t} \xrightarrow{op} B$ implies $m = 0$ and $n_i \geqslant 1$ for at least one i.

By $\sum_{i=1}^{p} (n_i + 1) = 2 + t$ we obtain :

$$\log_2 (\sum_{i=1}^{p} 2^{n_i} - t) \leqslant \log_2 (1 + \sum_{i=1}^{p} (2^{n_i} - 1)) \leqslant \sum_{i=1}^{p} n_i.$$

<u>C23</u> : If $\xi, \eta \in B \in \mathfrak{B}$ such that $\xi, \eta \in \max B$ and at least one of these elements is a singleton, say ξ, then we choose B^* with $\xi < \eta$ and obtain C2.

All other cases belong to C3 :

If $\xi = \max S^o_{n_j}$, $\eta = \max S^o_{n_k}$, where $j \neq k$ and $1 \leqslant n_j \leqslant n_k$, then we choose

$\xi < \eta$ in B^* and apply MIR^s to the $s = n_j$ many elements of $B^*_o = S^o_{n_j} \setminus \{\xi\}$. The inequality $f(A|B) \leqslant f(A'|B') + 1$ is easily checked. Since

$$MIR(A_{m,t}) = \{A_{m,t-1}, A_{m-1,t}\} \subseteq < A_{m-1,t} >,$$

it is sufficient to discuss $A' = A_{m-s,t}$ for $A = A_{m,t}$.

The most difficult case is given by $\xi \notin \max B$; then we choose $\xi < \eta$ in B^* and remove $\xi \in \min B$, hence $A' = A_{m-1,t}$, $B' = B \setminus \{\xi\}$, and there is one particular j with the modified $n'_j = n_j - 1$, whereas all the other values remain unchanged, $n'_i = n_i$ for $i \neq j$. With respect to C3 we have to show (cf. (7.1)) :

$$\lceil \log_2 (\sum_{i=1}^{p} 2^{n_i} - 2^{n_j - 1} - t) \rceil + 1 \geqslant \lceil \log_2 (\sum_{i=1}^{p} 2^{n_i} - t) \rceil .$$

Putting $d := t - \sum_{\substack{i=1 \\ i \neq j}} 2^{n_i}$, this is equivalent to :

$$\lceil \log_2 (2^{n_j} - 2d) \rceil \geqslant \lceil \log_2 (2^{n_j} - d) \rceil,$$

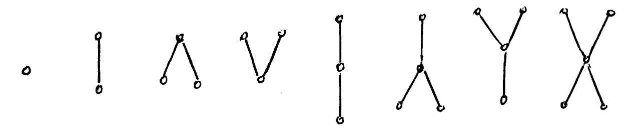
or to the simple condition $d < 2^{n_j - 2}$. Thus we need discuss only $d \geqslant 1$. From $\lambda(A|B) \geqslant 1$ and $n_j \geqslant 1$ we obtain $m \geqslant 1$, and (7.2) gives :

$$d \leqslant t - \sum_{\substack{i=1 \\ i \neq j}}^{p} (n_i + 1) = n_j + 1 - (m + 2),$$

$$n_j - 2 \geqslant d + m - 1 \geqslant d \geqslant 1 \ , \quad 2^{n_j - 2} > d.$$

8.-PROOF OF THEOREM 5.4

In order to show $\overline{\lambda}(S^2_2) \geqslant 6$ we choose the $\mathcal{G}$ of Theorem 6.4 as the smallest set of partial orders that contains $n\, S^2_2$ for all n and satisfies condition CO. Then each $A \in \mathcal{G}$ is a finite collection of pieces of type $P_1, \ldots, P_8$ that are given below together with associated weights w_i :

i	1	2	3	4	5	6	7	8
P_i								
w_i	0	1	2	2	3	4	4	6

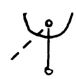
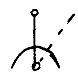
(8.1)

The other set $\mathcal{B}$ shall consist of all $B = t \cdot P_1 + m \cdot P_2$. For $A_K = k_1 P_1 + k_2 P_2 + \dots + k_8 P_8$, where $K = (k_1, \dots, k_8)$, and $B = t P_1 + m P_2$ with $t + m = \sum_i k_i$ we define

$$f(A_K | B) := \sum_{i=2}^{8} k_i w_i - m. \tag{8.2}$$

Condition C1 is satisfied, because $A_K \xrightarrow[\text{op}]{} t P_1 + m P_2$ implies $k_i = 0$ for $i \geqslant 3$ and $k_2 \leqslant m$.

If $\xi, \eta \in B$ are two singletons, then $B^* = (t-2)P_1 + (m+1)P_2$ satisfies condition C2.

The other cases lead to reductions  or 

These diagrams shall indicate that for ξ being the maximum of a P_2, we choose $\xi > \eta$ in B^* and apply MAR to ξ, and similarly, in the second case the dotted line shows our choice of B^* and the min-reduction . Each time we obtain $B' = (t+1)P_1 + (m-1)P_2$, thus m is reduced by 1.

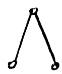
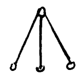
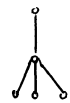
Applying MIR or MAR to A_K reduces $\sum_i k_i w_i$ by 2 at most, because of

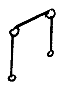
$$w_j \geqslant w_i - 2 \quad \text{for all } P_j \in \mathrm{MAR}(P_i) \cup \mathrm{MIR}(P_i). \tag{8.3}$$

Therefore, $f(A_K | B) = \sum_i k_i w_i - m$ can decrease by 1 at most.

After having checked all assuptions of Theorem 6.4 we apply (6.6) to $\mathcal{U} = \langle n P_i \rangle$, $B = n p_i \cdot T_1$ ($p_i = |P_i|$) and obtain $\lambda_o(n P_i) \geqslant n \cdot w_i$, but also $\lambda_o(P_i) = w_i$. This completes the proof of (5.9).

In order to prove $\overline{\lambda}(S_3^1) \geqslant 5.5$ we use a different $\mathcal{G} = \{A_K | K \in \mathbb{N}^7\}$, where now the P_i's and their weights are given by :

i	1	2	3	4	5	6	7	
P_i								(8.4)
w_i	0	1	2	2.5	3	4	5.5	

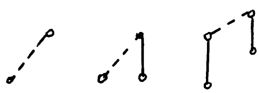
In addition we introduce P_o =  with weight $w_o = 3$. Then $\mathcal{B}$ is defined as the set of all

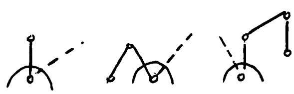
$$B_M = m_o P_o + m_1 P_1 + m_2 P_2 + m_3 P_3, \tag{8.5}$$

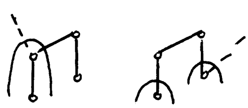
and the function f by :

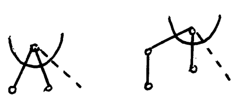
$$f(A_K | B_M) := \sum_{i=1}^{7} k_i w_i - \sum_{i=0}^{3} m_i w_i . \tag{8.6}$$

Assuming $A_K \xrightarrow[op]{} B_M$ condition C1 is checked by observing $k_6 = k_7 = 0$ and the fact that (ignoring singletons P_1) in P_2 only P_2 can be embedded, in P_3 only P_2 or P_3, and in P_o only P_3, P_4, P_5 or $2\,P_2$.

The C2 cases  increase $h = \sum_{o}^{3} m_i w_i$ by 1.

The MIR cases  decrease h by 1.

The MIR^2 cases 

and the MAR cases  decrease h by 2.

For checking C3 we need the additional bounds :

$$1.5 \geqslant w_i - w_j \quad \text{for} \quad P_j \in \mathrm{MIR}(P_i),$$
$$3 \geqslant w_i - w_j \quad \text{for} \quad P_j \in \mathrm{MIR}^2(P_i),$$
$$3 \geqslant w_i - w_j \quad \text{for} \quad P_j \in \mathrm{MAR}(P_i).$$

Finally, we apply (6.6) to $\mathcal{U} = < n P_7 >$, $B = 5 n P_1$, and obtain $\lambda_o(n S_3^1) / n \geqslant 5.5$.

***

REFERENCES

[1] FORD L.R. and JOHNSON S.M.: "A Tournament Problem", Amer. Math. Monthly 66, 387-389 (1959).

[2] HWANG F.K and LIN S. : "Optimal Merging of 2 Elements with n Elements" Acta Informatica 1, 145-158 (1971).

[3] KNUTH D.E. : "The Art of Computer Programming", vol. 3, (Addison-Wesley, 1973).

[4] PRATT V. and YAO F. : "On lower bounds for computing the i-th largest element", Proc. 14 th Ann. IEEE Symp. on Switching and Automata Theory, (1973) 70-81.

[5] SCHÖNHAGE A. , PATERSON M. and PIPPENGER N. : "Finding the Median" Theory of Comp. Report N°6 (April 1975), Univ. of Warwick.

Arnold SCHÖNHAGE
Mathematisches Institut
der Universität Tübingen
Auf der Morgenstelle 10
D-7400 TÜBINGEN 1 (R.F.A.)