Acta Informatica 1, 145–158 (1971)

Optimal Merging of 2 Elements with n Elements

F. K. HWANG and S. LIN

Received November 19, 1970

I. Introduction

Suppose we are given two disjoint linearly-ordered subsets A_m and B_n of a linearly-ordered set S, say

$$A_m = \{a_1 < a_2 < \cdots < a_m\}$$
$$B_n = \{b_1 < b_2 < \cdots < b_n\}.$$

The problem is to determine the linear ordering of their union (i.e., to merge A_m and B_n) by means of a sequence of pairwise comparisons between an element of A_m and an element of B_n. Given any algorithm s to solve this problem, which we refer to as the (m, n) problem, let $K_s(m, n)$ denote the maximum number of comparisons required and $E_s(m, n)$ the expected number of comparisons required to merge A_m and B_n using s, assuming that all possible initial orderings of $A_m \cup B_n$ are equally likely. An algorithm s is said to be M-optimal if $K_s(m, n) = K(m, n)$, where $K(m, n) = \min_x K_x(m, n)$, and E-optimal if $E_s(m, n) = E(m, n)$, where $E(m, n) = \min_x E_x(m, n)$.

In this paper, we construct an M-optimal algorithm to solve the $(2, n)$ problem and further show that this algorithm is E-optimal for infinitely many values of n.

II. Preliminary Results and Notations

Let $\mathscr{D}_0$ be the set of all possible orderings of $A_m \cup B_n$, and let $\mathscr{D}_k$ be the subset of $\mathscr{D}_0$ consistent with the results of the first k comparisons we have made thus far. Let G be the configuration consisting of A_m and B_n together with the set of relations R of the type $a_i > b_j$ or $a_i < b_j$ known as the result of these comparisons made. We shall refer to $\mathscr{D}_k$ as the set of data points associated with G. Let d_k denote the number of elements in $\mathscr{D}_k$. It is clear that, after making the i-th comparison ($i = 1, 2, \ldots, k$), one of the two possible outcomes must have $d_i \geqq \frac{1}{2} d_{i-1}$ and that merging is achieved if and only if $d_k = 1$. Since $d_0 = \binom{m+n}{m}$, we must have, for any algorithm s, $K_s(m, n) \geqq \left\lceil \log_2 \binom{m+n}{m} \right\rceil$*. It has also been established that if $d_0 = 2^\alpha + \theta$, $\alpha = \lfloor \log_2 d_0 \rfloor$, then $E_s(m, n) \geqq \alpha + \frac{2\theta}{d_0}$ and that this bound is achieved if and only if, after every comparison, the cardinalities of the two sets of data points associated with the two possible outcomes do not straddle an integral

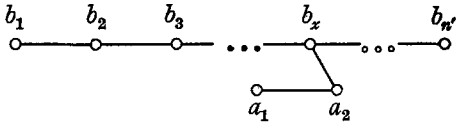
* As usual, we let $\lceil x \rceil$ denote the smallest integer $\geqq x$ and $\lfloor x \rfloor$ the largest integer $\leqq x$.

power of two [1]. Equivalently, one can see also that this lower bound of $\alpha + \frac{2\theta}{d_0}$ is achieved for $E_s(m, n)$ if and only if, for any data point in $\mathscr{D}_0$, algorithm s requires at least $K_s(m, n) - 1$ comparisons to identify it. (Hence exactly 2θ data points in $\mathscr{D}_0$ require $\alpha + \left\lceil \frac{\theta}{d_0} \right\rceil$ comparisons and $2^\alpha - \theta$ data points require α comparisons.) These bounds are usually referred to as information theory bounds.

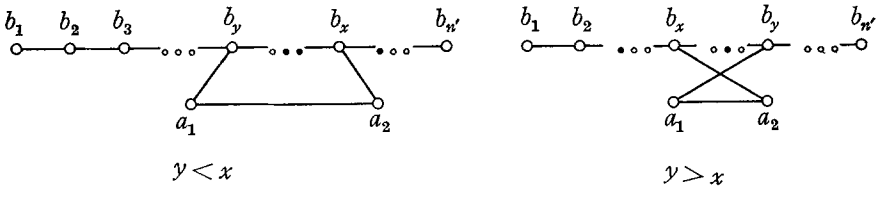
When $m = 1$ the merging problem is equivalent to the insertion problem. It is easy to see that the binary insertion algorithm, where we first compare a_1 with the "middle" element $b_{\left\lfloor \frac{n+1}{2} \right\rfloor}$ of the string in which it is to be inserted, achieves both the lower bounds for $K(1, n)$ and $E(1, n)$.

In the process of merging A_2 with B_n, the following configurations arise which we define below:

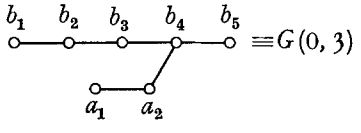
(1) $G(x, n') \equiv A_2(a_1 < a_2), B_{n'}(b_1 < b_2 < \cdots < b_{n'})$, together with the relation $a_2 > b_x$, $0 < x \leqq n'$. We also let $G(0, n')$ denote the configuration A_2 and $B_{n'}$ with no further relations.



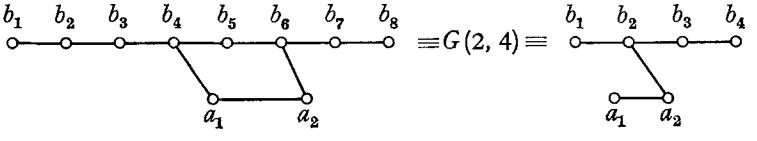
(2) $H(y, x, n') \equiv G(x, n')$, together with the relation $a_1 < b_y$, $1 \leqq y \leqq n'$. (Here y may be $\geqq$ or $< x$.)



Two configurations are said to be equivalent ($\equiv$) if they have isomorphic sets of data points. For example,



and



It is clear that for our purpose of merging, equivalent configurations can be used interchangeably with proper relabeling of the subscripts. For example, if we make

the comparison a_1 versus b_4 from the configuration $G(6, 8)$ and find that the outcome of the comparsion is $a_1 > b_4$, we shall say that the resulting configuration is $G(2, 4)$.

A configuration G_1 is said to dominate another configuration G_2 (written $G_1 > G_2$), if the set of data points belonging to G_2 is isomorphic to a subset of data points belonging to G_1. For example, $G(x, n) > G(y, n)$ if $x < y$. This can be seen from the fact that in $G(y, n)$, $a_2 > b_y$ implies $a_2 > b_y > b_x$. Similarly, $H(y, x, n) > H(y, y-1, n)$ if $y > x$.

Let $\hat{G}_s$ denote the maximum number of comparisons required to determine the linear ordering of the configuration G using algorithm s, and let $\hat{G} = \min_s \hat{G}_s$. It is clear that we have the following

Lemma 2.1.

(1) $G_1 \equiv G_2 \Rightarrow \hat{G}_1 = \hat{G}_2$.

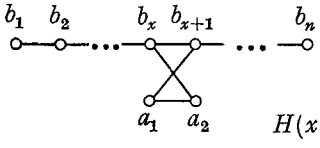
(2) $G_1 > G_2 \Rightarrow \hat{G}_1 \geqq \hat{G}_2$.

(3) $K(2, n) = \hat{G}(0, n)$.

(4) If s is any algorithm that produces the configuration G from $G(0, n)$ after t comparisons, then $K_s(2, n) \geqq t + \hat{G}$.

(5) If $G_1, \ldots, G_l$ are all the configurations that can arise from $G(0, n)$ after t comparisons following an algorithm s, then $K(2, n) \leqq K_s(2, n) \leqq t + \max_{1 \leqq i \leqq l} \hat{G}_i$.

Lemma 2.2. $\hat{H}(x+1, x, n) = \lceil \log_2(1+x) \rceil + \lceil \log_2(1+n-x) \rceil$.



$H(x+1, x, n)$

Proof. This is clear since a_1 and a_2 have to be merged independently into $b_1 < b_2 < \cdots < b_x$ and $b_{x+1} < \cdots < b_n$ respectively and binary insertion is the best way to achieve this.

Since $G(x, n) > H(x+1, x, n)$. we have

Corollary 2.2. $\hat{G}(x, n) \geqq \lceil \log_2(1+x) \rceil + \lceil \log_2(1+n-x) \rceil$.

Lemma 2.3. $\hat{G}(x, n) \leqq \lceil \log_2(1+n-x) \rceil + \lceil \log_2(1+n) \rceil$. $\hat{H}(y, x, n) \leqq \lceil \log_2 y \rceil + \lceil \log_2(1+n-x) \rceil$.

Proof. The algorithm to achieve this is obviously to merge a_2 into the string $b_{1+x} < \cdots < b_n$ by binary insertion and likewise a_1 next into the resulting string by binary insertion.

Since our work in the following will be constructive in nature, we shall henceforth assume that whenever Lemma 2.3 is used to give $\hat{G}(x, n) \leqq t$ or $\hat{H}(y, x, n) \leqq t$, we have in mind that we are using this particular algorithm.

Given any configuration G, let $G(a_i > b_j)$ denote the configuration G together with the additional relation $a_i > b_j$. Similarly for $G(a_i < b_j)$. The following lemma is of central importance in the derivation of $K(2, n)$ in the next section.

Lemma 2.4. Let G be any configuration arising from the merging of A_2 and B_n. If there exist z_1 and z_2 such that for $i=1$ and 2,

$$\hat{G}(a_i > b_{z_i}) \geqq k$$
$$\hat{G}(a_i < b_{1+z_i}) \geqq k,$$

then $\hat{G} \geqq 1+k$.

Proof. Let s be any M-optimal algorithm to linearize G whose first comparison involves a_i versus some b_x. If $x \leqq z_i$, let the outcome of the comparison be $a_i > v_x$. Then

$$\hat{G} \geqq 1+\hat{G}(a_i > b_x) \geqq 1+\hat{G}(a_i > b_{z_i}) \geqq 1+k,$$

since $a_i > b_{z_i}$ implies $a_i > b_x$. Similarly, if $x \geqq 1+z_i$, let the outcome of the comparison be $a_i < b_x$.

III. Determination of $K(2, n)$*

Let $T_i = i-1$ for $i=1, 2, 3, 4$, and for $k \geqq 3$ let

$$T_{2k-1} = T_{2k-2} + 2^{k-2}$$
$$T_{2k} = T_{2k-5} + 2^k. \qquad (R)$$

Some initial values of T_i are as follows:

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14
T_i	0	1	2	3	5	8	12	18	26	37	53	76	108	154

The main result in this section will be to show that if $T_{i-1} < n \leqq T_i$, then $K(2, n) = i$. This is accomplished by showing $K(2, T_i) \leqq i$ and $K(2, 1+T_i) \geqq 1+i$. Since the T_i's are monotonely increasing with i, and $K(2, n)$ is a nondecreasing function of n, the main result follows.

Before we prove $K(2, T_i) \leqq i$ and $K(2, 1+T_i) \geqq 1+i$, we derive an explicit expression for T_i from the recurrence formula (R).

Lemma 3.1.

$$1+T_{2k-1} = \left\lfloor (2^{k-1}+2^{k-2}) \sum_0^\infty 2^{-3i} \right\rfloor = \left\lfloor \frac{2^{k+2}+2^{k+1}}{7} \right\rfloor = \left\lfloor \frac{12}{7} 2^{k-1} \right\rfloor.$$

$$1+T_{2k} = \left\lfloor (2^k+2^{k-4}) \sum_0^\infty 2^{-3i} \right\rfloor = \left\lfloor \frac{2^{k+3}+2^{k-1}}{7} \right\rfloor = \left\lfloor \frac{17}{7} 2^{k-1} \right\rfloor.$$

Proof. Lemma 3.1 can easily be verified for $k=1$ and 2. The general expressions for $1+T_{2k-1}$ and $1+T_{2k}$ follow from using the recurrence formula (R). However,

* $K(2, n)$ has also been determined independently by Graham, using an entirely different approach [2]. The nice result for the explicit solution of T_i in Lemma 3.1 was suggested by his results, although Lemma 3.1 is not essential for our determination of $K(2, n)$ by constructive methods.

an inductive argument is more concise and we present it below. For $k \geqq 3$, we have

$$\begin{aligned} 1+T_{2k-1} &= 1+T_{2k-2}+2^{k-2} \\ &= \left\lfloor \frac{17}{7}\, 2^{k-2} \right\rfloor + 2^{k-2} \\ &= \left\lfloor \frac{12}{7}\, 2^{k-1} \right\rfloor . \end{aligned}$$

$$\begin{aligned} 1+T_{2k} &= 1+T_{2k-5}+2^{k} \\ &= \left\lfloor \frac{12}{7}\, 2^{k-3} \right\rfloor + 2^{k} \\ &= \left\lfloor \frac{68}{7}\, 2^{k-3} \right\rfloor \\ &= \left\lfloor \frac{17}{7}\, 2^{k-1} \right\rfloor . \end{aligned}$$

Lemma 3.2. For $k \geqq 3$, $T_{2k}-T_{2k-1} \leqq 2^{k-2}+2^{k-3}$.

Proof. We proceed by induction. Lemma 3.2 can easily be verified for $k=3$, 4, and 5. For $k \geqq 6$,

$$\begin{aligned} T_{2k}-T_{2k-1} &= T_{2k-5}+2^{k}-(T_{2k-2}+2^{k-2}) \\ &= T_{2k-6}+2^{k-4}+2^{k}-(T_{2k-7}+2^{k-1}+2^{k-2}) \\ &= T_{2k-6}-T_{2k-7}+2^{k-2}+2^{k-4} \\ &\leqq 2^{k-5}+2^{k-6}+2^{k-2}+2^{k-4} \leqq 2^{k-2}+2^{k-3}. \end{aligned}$$

In general, we may write (say for $k \geqq 4$)

$$1+T_{2k-1} = 2^{k-1}+2^{k-2}+2^{k-4}+\cdots$$

$$1+T_{2k} = 2^{k}+2^{k-3}+2^{k-4}+\cdots$$

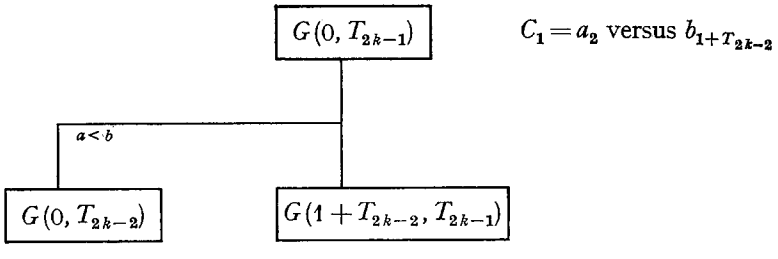
where $+\cdots$ means plus possibly some other powers of two whose sum is less than the last term indicated. We use this convention later in the paper to obtain bounds for the magnitudes of $1+T_{2k-1}$ and $1+T_{2k}$.

Theorem 3.1. $K(2, T_i) \leqq i$.

Proof. Theorem 3.1 can easily be verified for $i \leqq 4$. We proceed by induction, assuming that we have an algorithm to merge A_2 and B_{T_j} in no more than j comparisons for all $j < i$. Note that the induction is constructive in nature.

Schematically, we present the algorithm in the following two diagrams where each box contains the description of the configuration resulting from the comparison made in the preceding step. (C_j denotes the j^{th} comparison.)

Diagram I. $i = 2k-1$, $k \geqq 3$.



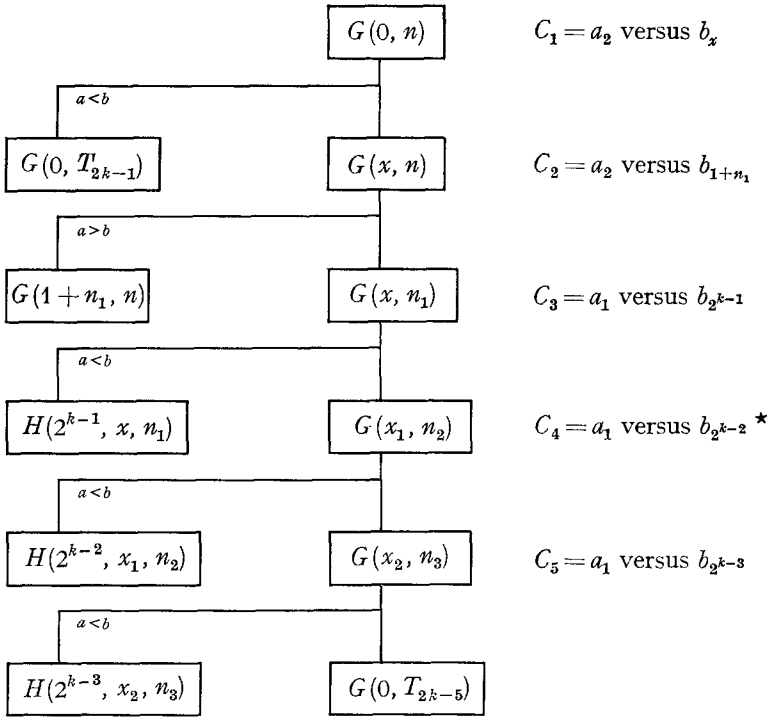
We have $\hat{G}(0, T_{2k-2}) \leqq 2k-2$ by induction; and by Lemma 2.3

$$\begin{aligned}\hat{G}(1+T_{2k-2}, T_{2k-1}) &\leqq \lceil \log_2(T_{2k-1}-T_{2k-2})\rceil + \lceil \log_2(1+T_{2k-1})\rceil \\ &\leqq (k-2)+k \\ &= 2k-2\end{aligned}$$

since we have $T_{2k-1}-T_{2k-2}=2^{k-2}$ and $1+T_{2k-1}<2^k$. Hence $\hat{G}(0, T_{2k-1}) \leqq 2k-1$.

Diagram II. $i=2k$, $k \geqq 3$. Let

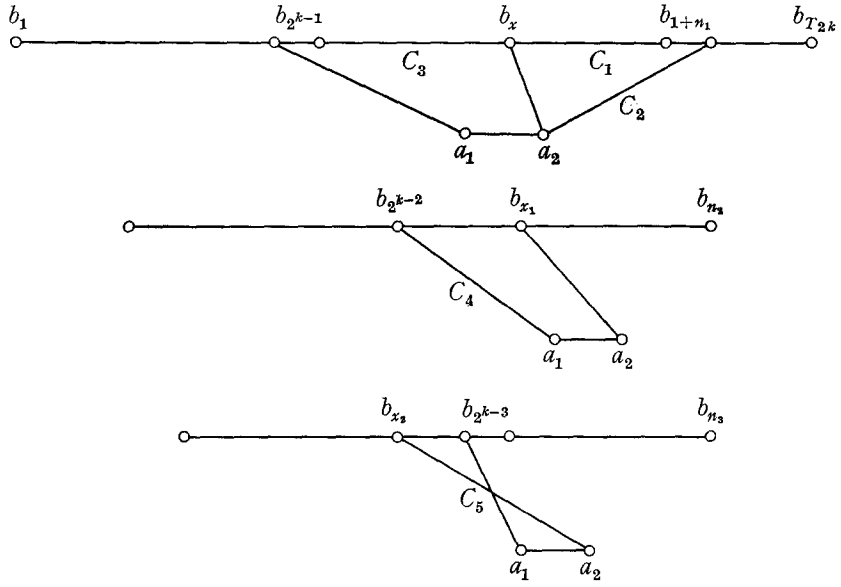
$$\begin{aligned} n &= T_{2k} \\ n_1 &= T_{2k}-2^{k-3} \\ n_2 &= T_{2k}-2^{k-3}-2^{k-1} \\ n_3 &= T_{2k}-2^{k-1}-2^{k-2}-2^{k-3} \\ x &= 1+T_{2k-1} \\ x_1 &= 1+T_{2k-1}-2^{k-1} \\ x_2 &= 1+T_{2k-1}-2^{k-1}-2^{k-2}. \end{aligned}$$



In the above diagram, we show the two configurations that arise after each comparison and put the more difficult case in the main line with further indication

* The subscript in b refers to the configuration in question, not necessarily to the original or previous configurations.

as to what comparison to perform next, while the other easier case is handled by a brief explanation below. The following figures illustrate graphically the configuration in the main line after each comparison.



We have: $1+n_1-x=1+n_2-x_1=1+n_3-x_2=T_{2k}-T_{2k-1}-2^{k-3}\leqq 2^{k-2}$ by Lemma 3.2 and $1+n=1+T_{2k}<2^{k+1}$. Furthermore,

$$\hat{G}(0,\ T_{2k-1})\leqq 2k-1$$

by induction, and

$$\begin{aligned}\hat{G}(1+n_1,\ n) &\leqq \lceil\log_2(n-n_1)\rceil+\lceil\log_2(1+n)\rceil\\ &\leqq k-3+(k+1)=2k-2,\end{aligned}$$

$$\begin{aligned}\hat{H}(2^{k-1},\ x,\ n_1) &\leqq k-1+\lceil\log_2(1+n_1-x)\rceil\\ &\leqq k-1+(k-2)=2k-3,\end{aligned}$$

$$\begin{aligned}\hat{H}(2^{k-2},\ x_1,\ n_2) &\leqq k-2+\lceil\log_2(1+n_2-x_1)\rceil\\ &\leqq k-2+(k-2)=2k-4,\end{aligned}$$

$$\begin{aligned}\hat{H}(2^{k-3},\ x_2,\ n_3) &\leqq k-3+\lceil\log_2(1+n_3-x_2)\rceil\\ &\leqq k-3+(k-2)=2k-5,\end{aligned}$$

by Lemma 2.3.

Since $n_3-2^{k-3}=T_{2k}-2^k=T_{2k-5}$, and $x_2=1+T_{2k-1}-2^{k-1}-2^{k-2}<2^{k-3}$, we see that if the outcome of C_5 is $a_1>b_{2^{k-3}}$, the resulting configuration is $G(0,\ T_{2k-5})$ with $\hat{G}(0,\ T_{2k-5})\leqq 2k-5$ by induction. Hence $\hat{G}(0,\ T_{2k})\leqq 2k$, and Theorem 3.1 is proved.

Theorem 3.2. $K(2,\ 1+T_i)\geqq 1+i$.

Proof. Theorem 3.2 can easily be verified for $i \leqq 4$. We proceed by induction, assuming that $K(2, 1+T_j) \geqq 1+j$ for all $j<i$. Let s be any algorithm to do the $(2, 1+T_i)$ problem. By symmetry, we may assume that the first comparison C_1 is a_2 versus some b_x.

If $x>1+T_{i-1}$, let the outcome of the comparison be $a_2<b_x$, with resulting configuration $G(0, x-1)$. Since $\hat{G}(0, x-1) \geqq \hat{G}(0, 1+T_{i-1}) \geqq i$ by induction, $K_s(2, 1+T_i) \geqq 1+i$.

If $x \leqq 1+T_{i-1}$, let the outcome of the comparison be $a_2>b_x$, with resulting configuration $G(x, 1+T_i)$. Since $\hat{G}(x, 1+T_i) \geqq \hat{G}(1+T_{i-1}, 1+T_i)$, the theorem will be proved if we can show $\hat{G}(1+T_{i-1}, 1+T_i) \geqq i$. We divide the remaining work into two cases:

Case 1. $i=2k-1$. From Corollary 2.2, we have

$$\hat{G}(1+T_{i-1}, 1+T_i) \geqq \lceil \log_2(1+T_i-T_{i-1}) \rceil + \lceil \log_2(2+T_{i-1}) \rceil.$$

Since $1+T_i-T_{i-1}=1+T_{2k-1}-T_{2k-2}=1+2^{k-2}$ and $2+T_{i-1}=2+T_{2k-2}>2^{k-1}$, we have

$$\hat{G}(1+T_{i-1}, 1+T_i) \geqq (k-1)+k=i.$$

Case 2. $i=2k$. This case is more difficult because Corollary 2.2 does not give the desired results. It can be verified that the number of data points satisfying the configuration $G(1+T_5, 1+T_6)$ is 34 and hence $\hat{G}(1+T_5, 1+T_6) \geqq \lceil \log_2 34 \rceil = 6$. Therefore, we shall assume in the following that $k \geqq 4$.

We proceed by proving the following:

(1) Let $\alpha = 1+T_{2k-1}-2^{k-1}-2^{k-2}=2^{k-4}+\cdots$,
$\beta = 1+T_{2k}-2^{k-1}-2^{k-2}-2^{k-3}=2^{k-2}+2^{k-4}+\cdots$,
then $\hat{G}(\alpha, \beta) \geqq 2k-3$.

Proof. Applying Lemma 2.4, let $z_1=2^{k-3}$, $z_2=\beta-2^{k-4}$. We have $\alpha<z_1=2^{k-3}$, $z_2 \geqq 2^{k-2}$, $\beta-z_1 \geqq 2^{k-3}$, and (with the help of Lemma 2.2 and its corollary)

$$\begin{aligned}
\hat{G}(a_1<b_{1-z_1}) &\geqq \hat{H}(1+z_1, z_1, \beta) \\
&= \lceil \log_2(1+z_1) \rceil + \lceil \log_2(1+\beta-z_1) \rceil \\
&\geqq (k-2)+(k-2) \\
&= 2k-4,
\end{aligned}$$

$$\begin{aligned}
\hat{G}(a_1>b_{z_1}) &= \hat{G}(0, \beta-z_1) \\
&= \hat{G}(0, 1+T_{2k}-2^k) \\
&= \hat{G}(0, 1+T_{2k-5}) \geqq 2k-4 \quad \text{by induction,}
\end{aligned}$$

$$\begin{aligned}
\hat{G}(a_2<b_{1+z_1}) &= \hat{G}(\alpha, z_2) \\
&\geqq \hat{G}(2^{k-3}, 2^{k-2}) \\
&\geqq \lceil \log_2(1+2^{k-3}) \rceil + \lceil \log_2(1+2^{k-2}-2^{k-3}) \rceil \\
&= 2k-4,
\end{aligned}$$

$$\begin{aligned}\hat{G}(a_2 > b_{z_2}) &= \hat{G}(z_2, \beta)\\ &\geqq \lceil \log_2(1+z_2)\rceil + \lceil \log_2(1+\beta-z_2)\rceil\\ &\geqq (k-1)+(k-3)\\ &= 2k-4.\end{aligned}$$

Hence $\hat{G}(1+T_{2k-1}-2^{k-1}-2^{k-2},\ 1+T_{2k}-2^{k-1}-2^{k-2}-2^{k-3}) \geqq 2k-3$.

(2) Let $\alpha = 1+T_{2k-1}-2^{k-1} = 2^{k-2}+2^{k-4}+\cdots$

$\beta = 1+T_{2k}-2^{k-1}-2^{k-3} = 2^{k-1}+2^{k-4}+\cdots,$

then $\hat{G}(\alpha, \beta) \geqq 2k-2$.

Proof. Applying Lemma 2.4, let $z_1 = 2^{k-2}$, $z_2 = \beta - 2^{k-4}$. We have $\beta - \alpha > z_2 - \alpha > 2^{k-3}$, and

$$\begin{aligned}\hat{G}(a_1 < b_{1+z_1}) &\geqq \lceil \log_2(1+z_1)\rceil + \lceil \log_2(1+\beta-\alpha)\rceil\\ &\geqq (k-1)+(k-2)\\ &= 2k-3,\end{aligned}$$

$$\begin{aligned}\hat{G}(a_1 > b_{z_1}) &= \hat{G}(\alpha-2^{k-2}, \beta-2^{k-2})\\ &\geqq 2k-3 \quad \text{by (1)},\end{aligned}$$

$$\begin{aligned}\hat{G}(a_2 < b_{1+z_2}) &= \hat{G}(\alpha, z_2)\\ &\geqq \lceil \log_2(1+\alpha)\rceil + \lceil \log_2(1+z_2-\alpha)\rceil\\ &\geqq (k-1)+(k-2)\\ &= 2k-3,\end{aligned}$$

$$\begin{aligned}\hat{G}(a_2 > b_{z_2}) &= \hat{G}(z_2, \beta)\\ &\geqq \lceil \log_2(1+z_2)\rceil + \lceil \log_2(1+\beta-z_2)\rceil\\ &\geqq k+k-3\\ &= 2k-3.\end{aligned}$$

Hence $\hat{G}(1+T_{2k-1}-2^{k-1},\ 1+T_{2k}-2^{k-1}-2^{k-3}) \geqq 2k-2$.

(2*) Let $\alpha = 1+T_{2k-1}-2^{k-1}-2^{k-2} = 2^{k-4}+\cdots$

$\beta = 1+T_{2k}-2^{k-1}-2^{k-2} = 2^{k-2}+2^{k-3}+2^{k-4}+\cdots,$

then $\hat{G}(\alpha, \beta) \geqq 2k-2$.

Proof. Applying Lemma 2.4, let $z_1 = 2^{k-3}$, $z_2 = \beta - 2^{k-3}$. We have $\alpha < z_1 = 2^{k-3}$, $z_2 \geqq 2^{k-2}$, $\beta - z_1 \geqq 2^{k-2}+2^{k-4}+\cdots \geqq 1+T_{2k-4}$, and

$$\begin{aligned}\hat{G}(a_1 < b_{1+z_1}) &\geqq \hat{H}(1+z_1, z_1, \beta)\\ &\geqq \lceil \log_2(1+z_1)\rceil + \lceil \log_2(1+\beta-z_1)\rceil\\ &\geqq (k-2)+(k-1)\\ &= 2k-3,\end{aligned}$$

$$\begin{aligned}
\hat{G}(a_1 > b_{z_1}) &= \hat{G}(0, \beta - z_1) \\
&\geqq \hat{G}(0, 1 + T_{2k-4}) \\
&\geqq 2k-3 \quad \text{by induction,} \\
\hat{G}(a_2 < b_{1+z_2}) &= \hat{G}(\alpha, z_2) \\
&\geqq 2k-3 \quad \text{by (1),} \\
\hat{G}(a_2 > b_{z_2}) &= \hat{G}(z_2, \beta) \\
&\geqq \lceil \log_2(1+z_2) \rceil + \lceil \log_2(1+\beta-z_2) \rceil \\
&\geqq (k-1) + (k-2) \\
&= 2k-3.
\end{aligned}$$

Hence $\hat{G}(1+T_{2k-1}-2^{k-1}-2^{k-2},\ 1+T_{2k}-2^{k-1}-2^{k-2}) \geqq 2k-2$.

(3) Let $\alpha = 1+T_{2k-1} = 2^{k-1}+2^{k-2}+2^{k-4}+\cdots$,

$\beta = 1+T_{2k}-2^{k-3} = 2^k+2^{k-4}+\cdots$,

then $\hat{G}(\alpha, \beta) \geqq 2k-1$.

Proof. Applying Lemma 2.4, let $z_1 = 2^{k-1}$, $z_2 = \beta - 2^{k-4}$. We have

$$\beta - \alpha > z_2 - \alpha > 2^{k-3},$$

and

$$\begin{aligned}
\hat{G}(a_1 < b_{1+z_1}) &\geqq \lceil \log_2(1+z_1) \rceil + \lceil \log_2(1+\beta-\alpha) \rceil \\
&\geqq k + (k-2) \\
&= 2k-2, \\
\hat{G}(a_1 > b_{z_1}) &= \hat{G}(\alpha - 2^{k-1}, \beta - 2^{k-1}) \\
&\geqq 2k-2 \quad \text{by (2),} \\
\hat{G}(a_2 < b_{1+z_2}) &= \hat{G}(\alpha, z_2) \\
&\geqq \lceil \log_2(1+\alpha) \rceil + \lceil \log_2(1+z_2-\alpha) \rceil \\
&\geqq k + (k-2) \\
&= 2k-2, \\
\hat{G}(a_2 > b_{z_2}) &= \hat{G}(z_2, \beta) \\
&\geqq \lceil \log_2(1+z_2) \rceil + \lceil \log_2(1+\beta-z_2) \rceil \\
&\geqq (k+1) + (k-3) \\
&= 2k-2.
\end{aligned}$$

Hence $\hat{G}(1+T_{2k-1},\ 1+T_{2k}-2^{k-3}) \geqq 2k-1$.

(3*) Let $\alpha = 1+T_{2k-1}-2^{k-1} = 2^{k-2}+2^{k-4}+\cdots$

$\beta = 1+T_{2k}-2^{k-1} = 2^{k-1}+2^{k-3}+2^{k-4}+\cdots$,

then $\hat{G}(\alpha, \beta) \geqq 2k-1$.

Proof. Applying Lemma 2.4, let $z_1 = 2^{k-2}$, $z_2 = \beta - 2^{k-3}$. We have

$$\beta - \alpha > 2^{k-2},$$

and

$$\begin{aligned}\hat{G}(a_1<b_{1+z_1}) &\geqq \lceil\log_2(1+z_1)\rceil+\lceil\log_2(1+\beta-\alpha)\rceil\\ &\geqq (k-1)+(k-1)\\ &=2k-2,\end{aligned}$$

$$\begin{aligned}\hat{G}(a_1>b_{z_1}) &=\hat{G}(\alpha-z_1,\ \beta-z_1)\\ &\geqq 2k-2 \quad \text{by } (2^*)\end{aligned}$$

$$\begin{aligned}\hat{G}(a_2<b_{1+z_2}) &=\hat{G}(\alpha,\ z_2)\\ &\geqq 2k-2 \quad \text{by } (2)\end{aligned}$$

$$\begin{aligned}\hat{G}(a_2>b_{z_2}) &=\hat{G}(z_2,\ \beta)\\ &\geqq \lceil\log_2(1+z_2)\rceil+\lceil\log_2(1+\beta-z_2)\rceil\\ &\geqq k+(k-2)\\ &=2k-2.\end{aligned}$$

Hence $\hat{G}(1+T_{2k-1}-2^{k-1},\ 1+T_{2k}-2^{k-1})\geqq 2k-1$.

(4) Let $\alpha=1+T_{2k-1}=2^{k-1}+2^{k-2}+2^{k-4}+\cdots$,

$\beta=1+T_{2k}=2^k+2^{k-3}+2^{k-4}+\cdots$,

then $\hat{G}(\alpha,\beta)\geqq 2k$.

Proof. Applying Lemma 2.4, let $z_1=2^{k-1}$, $z_2=\beta-2^{k-3}$. We have $\beta-\alpha>2^{k-2}$ and

$$\begin{aligned}\hat{G}(a_1<b_{1+z_1}) &\geqq \lceil\log_2(1+z_1)\rceil+\lceil\log_2(1+\beta-\alpha)\rceil\\ &\geqq k+(k-1)\\ &=2k-1,\end{aligned}$$

$$\begin{aligned}\hat{G}(a_1>b_{z_1}) &=\hat{G}(\alpha-2^{k-1},\ \beta-2^{k-1})\\ &\geqq 2k-1 \quad \text{by } (3^*),\end{aligned}$$

$$\begin{aligned}\hat{G}(a_2<b_{1+z_2}) &=\hat{G}(\alpha,\ z_2)\\ &\geqq 2k-1 \quad \text{by } (3),\end{aligned}$$

$$\begin{aligned}\hat{G}(a_2>b_{z_2}) &=\hat{G}(z_2,\ \beta)\\ &\geqq \lceil\log_2(1+z_2)\rceil+\lceil\log_2(1+\beta-z_2)\rceil\\ &\geqq (k+1)+(k-2)\\ &=2k-1.\end{aligned}$$

Hence $\hat{G}(1+T_{2k-1},\ 1+T_{2k})\geqq 2k$, and Theorem 3.2 is proved.

Theorem 3.3. The algorithm indicated in Theorem 3.1 to solve the $(2, n)$ problem is also E-optimal for $n=T_i$.

Proof. We show the algorithm indicated in Theorem 3.1 requires at least $i-1$ comparisons to identify any data point in $\mathscr{D}_0$, and hence achieves the information-theoretic lower bound for the expected number of comparisons when $n=T_i$. Theorem 3.3 can easily be verified for $i \leqq 6$. We proceed by induction by assuming that for all $j<i$, the algorithm requires at least $j-1$ comparisons to identify any of its data points in solving the $(2, T_j)$ problem. Referring to Diagrams I and II in the proof of Theorem 3.1, we have:

Case I. $i=2k-1$, $k \geqq 4$.

The set of data points satisfying $a_2<b_{1+T_{2k-2}}$ after the first comparison requires at least $1+2k-3=i-1$ comparisons by induction. The set of data points satisfying $a_2>b_{1+T_{2k-2}}$ requires at least

$$1+k-2+\lfloor \log_2(2+T_{2k-2}) \rfloor = 1+k-2+k-1=i-1$$

comparisons as in the case when $b_{1+T_{2k-2}}<a_2<b_{2+T_{2k-2}}$.

Case II. $i=2k$. $k \geqq 4$.

(1) After C_1, the set of data points satisfying $a_2<b_{1+T_{2k-1}}$ requires at least $1+2k-2=i-1$ comparisons by induction.

(2) After C_2, the set of data points satisfying $a_2> b_{T_{2k}-2^{k-3}+1}$ requires at least $2+k-3+\lfloor \log(2+T_{2k}-2^{k-3}) \rfloor = 2+k-3+k=i-1$ comparisons as in the case when $b_{1+T_{2k}-2^{k-3}}<a_2<b_{2+T_{2k}-2^{k-3}}$.

(3) After C_3, the set of data points satisfying $a_1<b_{2^{k-1}}$ requires at least $3+k-1+\lfloor \log_2(T_{2k}-T_{2k+1}-2^{k-3}) \rfloor \geqq 3+k-1+k-3=i-1$ comparisons.

(4) After C_4, the set of data points satisfying $a_1<b_{2^{k-2}}$ requires at least $4+k-2+\lfloor \log_2(T_{2k}-T_{2k-1}-2^{k-3}) \rfloor \geqq 4+k-2+k-3=i-1$ comparisons.

(5) After C_5, the set of data points satisfying $a_1>b_{2^{k-3}}$ requires at least $5+(2k-6)=i-1$ comparisons by induction, and the set of data points satisfying $a_1<b_{2^{k-3}}$ requires at least $5+k-3+\lfloor \log_2(2+T_{2k-5}) \rfloor \geqq 5+k-3+k-3=i-1$ comparisons as in the case when $b_{2^{k-3}-1}<a_1<b_{2^{k-3}}$.

Hence Theorem 3.3 is proved.

Corollary 3.3. Let

$$d_0=\binom{T_i+2}{2}=2^\alpha+\theta$$

where $\alpha=\lfloor \log_2 d_0 \rfloor$. Then

$$K(2, T_i)=\alpha+\left\lfloor \frac{\theta}{d_0} \right\rfloor = i$$

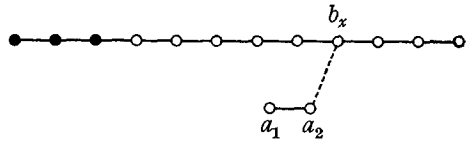
and

$$E(2, T_i)=\alpha+\frac{2\theta}{d_0}.$$

Although Theorem 3.1 does not specify what to do for values of n other than T_i, it is clear that if $T_i<n<T_{i+1}$, we may add $t=T_{i+1}-n$ dummy elements to the string of b's and proceed accordingly. For example, in solving the (2, 9)

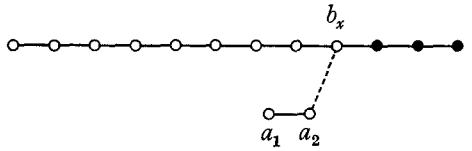
problem, the two extreme cases can happen: (Add three dummy elements)

(1)



• denotes dummy elements.

(2)



and we see that the first comparison can be a_2 versus b_x where $6 \leqq x \leqq 9$. Different choices of the values of x can affect the expected number of comparisons required. The determination of the value of x which will minimize the expected number of comparisons required, even assuming that we can proceed optimally thereafter remains a difficult problem. In this particular case, we have verified that either $x=6$ or 7 gives $E(2, 9)=5\frac{48}{55}$, which illustrates the interesting fact that an algorithm which is both M-optimal and E-optimal need not be unique. The following table lists the values of $E(2, n)$ for $n \leqq 10$.

n	$K(2, n)$	K-bound	$E(2, n)$	E-bound	First comparison to achieve $E(2, n)$
1	2	2	$\frac{5}{3}$	$\frac{5}{3}$	
2	3	3	$\frac{16}{6}$	$\frac{16}{6}$	
3	4	4	$\frac{34}{10}$	$\frac{34}{10}$	
4	5	4	$\frac{60}{15}$	$\frac{59}{15}$	a_2 versus b_3
5	5	5	$\frac{94}{21}$	$\frac{94}{21}$	
6	6	5	$\frac{137}{28}$	$\frac{136}{28}$	a_2 versus b_5
7	6	6	$\frac{189}{36}$	$\frac{188}{36}$	a_2 versus b_6
8	6	6	$\frac{251}{45}$	$\frac{251}{45}$	
9	7	6	$\frac{323}{55}$	$\frac{321}{55}$	a_2 versus b_6 or b_7
10	7	7	$\frac{405}{66}$	$\frac{400}{66}$	a_2 versus b_7

If $d_0=\binom{n+2}{2}=2^\alpha+\theta$ where $\alpha=\lfloor \log_2 d_0 \rfloor$, the information theory bounds for $K(2, n)$ and $E(2, n)$ are indicated by K-bound $=\alpha+\left\lceil \frac{\theta}{d_0} \right\rceil$ and E-bound $=\alpha+\frac{2\theta}{d_0}$ respectively. Note that even for $n=4$, neither bound can be achieved, while Corollary 3.3 states that both bounds can be simultaneously achieved for $n=T_i$ by the algorithm developed in this paper.

References

1. Sandelius, M.: On an optimal search procedure. Amer. Math. Monthly **68**, No. 2, 133–134.
2. Graham, R. L.: On sorting by comparisons. To appear in Proceedings of Atlas Symposium, No. 2.

F. K. Hwang
S. Lin
Bell Telephone Labs. Inc.
Murray Hill, New Jersey 07974
U.S.A.