JOURNAL OF COMPUTER AND SYSTEM SCIENCES 7, 615–621 (1973)

# Two Entropies of a Generalized Sorting Problem

AKIHIRO NOZAKI

*Information Science Laboratory, Faculty of Science, University of Tokyo*

Received September 19, 1972

In this paper we introduce a class of generalized sorting (ordering) problems called "classifications." To each "classification," we associate two quantities: *informational entropy* (average information quantity) and *operational entropy* (measure of computational complexity, that is, number of comparisons necessary to "classify" a given sequence of items). The relationship between these quantities is discussed. For a certain classification involving $n$ items, its operational entropy is shown to be approximately $n \cdot \log_2 n$ although its informational entropy is constantly equal to 1, independent of the number of items $n$.

## 1. INTRODUCTION

As it is well known, the complete sorting of $n$ items can be carried out in approximately ($\log_2 n!$) pairwise comparisons of their keys.[1] The number ($\log_2 n!$) is equal to the information quantity to distinguish a case among $n!$ equally possible cases. Thus the information quantity may seem to have close relationship to the "operational entropy," that is, the number of comparisons necessary to distinguish a case (cf. Burge [1]). However, this relationship is not always so close. For instance, in order to find the strongest among $n$ baseball teams, $n - 1$ comparisons (games) are necessary although the information quantity to know the strongest is at most $\log_2 n$.

In this paper, we consider a certain generalization of sorting problem which we call classification problem. We define for each "classification" its *operational entropy* as well as its *informational entropy* (average information quantity) and investigate the difference between these quantities.

[1] $\log_2 n! \doteq (\log_2 n - 1.443)(n + 1/2) \doteq n \log_2 n$. By binary merging (straight two-way merge), $n$ items can be sorted in $n \cdot \lceil \log_2 n \rceil$ times of comparisons (see, for instance, [2]). If we employ binary insertion,

$$\sum_{i=1}^{n} \lceil \log_2 i \rceil$$

times of comparisons are sufficient (see [2]). A more efficient algorithm was found by Ford–Johnson [4].

## 2. Classification

We shall start by giving a typical example of the classification problem.

*Given n distinct real numbers* $a_1, \ldots, a_n$, *find the least number by repeating pairwise comparisons.*

Suppose that

$$a_{i(1)} < a_{i(2)} < \cdots \cdots < a_{i(n)} .$$

We define a permutation $\tau$ of $n$ integers as follows:

$$\tau = \begin{pmatrix} i(1), & i(2), & \ldots, & i(n) \\ 1, & 2, & \ldots, & n \end{pmatrix}.$$

Thus the number $a_j$ is the $\tau(j)$-th least number of $a_i$'s.

Now let $C_i$ be the set of all permutations which map $i$ to 1, that is,

$$C_i = \{\sigma \in S_n \,;\, \sigma(i) = 1\},$$

where $S_n$ is the set of all permutations of integers $1, \ldots, n$. The aim of the above mentioned problem is then stated as follows: given a permutation $\tau$, find a set $C_i$ such that

$$\tau \in C_i .$$

Another problem will be introduced by changing the definition of $C_i$'s. Such a problem is generally represented by a classification defined as following.

Definition 1. Let $S_n$ be the whole set of permutations of $n$ integers $1, \ldots, n$.

A classification $\mathscr{C}$ in $S_n$ is a partition of a subset of $S_n$, that is, a set of disjoint subsets of $S_n$.

The aim of a "classification problem" is to "classify" a given permutation $\tau$ in $\mathscr{C}$, that is, to find the set $C$ in $\mathscr{C}$ which contains $\tau$.

*Remark.* The goal of the complete sorting is represented by the following classification $\mathscr{S}(n)$:

$$\mathscr{S}(n) = \{\{\sigma\}; \sigma \in S_n\}.$$

Definition 2. Let

$$\mathscr{C} = \{C_i \,;\, 1 \leqslant i \leqslant t\}$$

be a classification in $S_n$.

(1) The *operational entropy* $q(\mathscr{C})$ of the classification $\mathscr{C}$ is the worst-case number of comparisons required to classify a given permutation in $\mathscr{C}$.

(2) The *informational entropy* $e(\mathscr{C})$ of the classification $\mathscr{C}$ is the average information quantity of the events "$\tau \in C_1$",..., "$\tau \in C_t$" whose probabilities are proportional to the cardinalities of $C_1, \ldots, C_t$, respectively. More precisely,

$$P(\tau \in C_i) = \frac{\text{the number of permutations in } C_i}{\text{the total number of permutations in } \mathscr{C}}.$$

*Remark.* If every set $C_i$ contains the same number of permutations, then $e(\mathscr{C}) = \log_2 t$.

In what follows, we shall consider the relationship between these entropies, operational and informational.

## 3. Entropies: Operational and Informational

Let $\mathscr{S}(n)$ be the classification representing the complete sorting. We denote by $S(n)$ its operational entropy $q(\mathscr{S}(n))$. Then the following propositions are immediate.

Proposition 1. *For any classification $\mathscr{C}$ in $S_n$,*

$$S(n) \geqslant q(\mathscr{C}) \geqslant e(\mathscr{C}).$$

Proposition 2.

$$S(n) \doteq e(\mathscr{S}(n)) \doteq n \log_2 n.$$

Thus the classification $\mathscr{S}(n)$ is an extreme case in which two entropies $q(\mathscr{S}(n))$ and $e(\mathscr{S}(n))$ have no significant difference. An opposite extreme in this regard is the classification defined as follows:

$$\mathscr{P}(n) = \{S_n - A_n, A_n\},$$

where $A_n$ is the set of all even permutations in $S_n$. This classification obviously corresponds to the determination of the parity of a given permutation.

Theorem 1.

(1) $e(\mathscr{P}(n)) = 1$,

(2) $q(\mathscr{P}(n)) = S(n)$.

*Proof.* (1) The property (1) is obvious.

(2) By Proposition 1,

$$q(\mathscr{P}(n)) \leqslant S(n).$$

We shall therefore show the reversed inequality.

Suppose that we can tell whether a given permutation $\tau$ is even or odd, after obtaining the results of $p$ comparisons, say: $\tau(i_1) < \tau(j_1),\dots, \tau(i_p) < \tau(j_p)$. By these relations we can draw a Hasse diagram which shows a part of the order: $\tau^{-1}(1), \tau^{-1}(2),\dots, \tau^{-1}(n)$ (see Fig. 1). We denote by $h(i)$ the "height" of the node $i$ in this diagram which is precisely defined as follows:

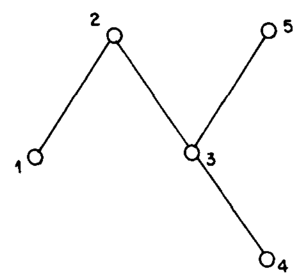


FIG. 1. A diagram representing the results of comparisons as follows: $a_1 < a_2$, $a_4 < a_3$, $a_3 < a_2$ and $a_3 < a_5$. In this case, $h(1) = h(4) = 1$, $h(3) = 2$ and $h(2) = h(5) = 3$.

(i) If there is no integer $s$ such that $j_s = i$, then

$$h(i) = 1.$$

(ii) Otherwise,

$$h(i) = 1 + \max\{h(i_s); j_s = i\}.$$

We shall see that $h(i) \neq h(j)$ for any distinct integers $i$ and $j$.

Let $R$ be the set of all permutations satisfying the following condition.

$$\textit{If } h(i) < h(j), \textit{ then } \sigma(i) < \sigma(j). \tag{1}$$

Such permutations, $\sigma$'s, are either all even or all odd since, if not, we can not tell whether the given permutation $\tau$ ($\in R$) is even or odd.

Suppose that there are distinct integers $i$ and $j$ such that $h(i) = h(j)$. Then the permutation

$$\tau' = \tau \cdot [i, j]$$

also satisfies condition (1) and is in $R$, where $[i, j]$ denotes the transposition which exchanges $i$ and $j$. But this is not the case since $R$ can not contain both $\tau$ and $\tau'$, one is even and the other is odd.

The mapping $h$ is therefore one-to-one. The diagram representing the results of comparisons is then linear and we can tell what $\tau$ is:

$$\tau = \begin{pmatrix} 1, & 1, & \dots, & n \\ h(1), & h(2), & \dots, & h(n) \end{pmatrix}.$$

Whenever we know whether $\tau$ is even or not, we can tell what $\tau$ is. Therefore,

$$q(\mathscr{P}(n)) \geqslant q(\mathscr{S}(n)) = S(n).$$

This completes the proof of the Theorem 1.

Now let us consider another interesting example, the classification $\mathscr{Y}(k)$ defined as follows:

$$C(i_1, \ldots, i_k) = \{\sigma \in S_{2k} ; \sigma(\{i_1, \ldots, i_k\}) \subseteq \{i_1, \ldots, i_k\}\},$$
$$\mathscr{Y}(k) = \{C(i_1, \ldots, i_k); 1 \leqslant i_1 < \cdots < i_k \leqslant 2k\}.$$

This classification represents the selection of the least $k$ ones among $2k$ distinct real numbers (see Yoneda [4]). We denote:

$$Y(k) = q(\mathscr{Y}(k)),$$
$$y(k) = \lceil e(\mathscr{Y}(k))\rceil.$$

Exact evaluation of $Y(k)$ is at present an open problem. We can nevertheless give some upper and lower bounds and see that $6 > Y(k)/y(k) > 1$ for large $k$.

THEOREM 2.

(1) $y(k) = \lceil \log_2({}_{2k}C_k)\rceil \doteq 2k - (1/2)\log_2 k$

(2) $10.87k \geqslant Y(k) \geqslant 3k - 2.$

*Remark* 1. The upper bound of $Y(k)$ is due to M. Blum *et al.* ([5]). In fact, the value $10.87k$ is an upper bound for a much harder problem (the median computation.) They gave also a slightly weaker lower bound for the harder problem, which yields $Y(k) \geqslant 3k - 6$.

*Remark* 2. Ikeno and Simauti gave another upper bound of $Y(k)$ as follows ([7]):

$$Y(k) \leqslant 2 \cdot S(k) + \lfloor k/2 \rfloor + 1.$$

Although it is not linear, this bound is better than (2) for small values of $k$ ($k \leqslant 36$).

*Proof.* We shall only show that

$$Y(k) \geqslant 3k - 2. \tag{2}$$

It is easy to see that $Y(1) = 1$ and $Y(2) = 4$. Thus inequality (2) is true for $k = 1$ and 2. We shall therefore show that

$$Y(k+1) - 3 \geqslant Y(k) \tag{3}$$

for $k \geqslant 2$, by illustrating an algorithm to find the least $k$ ones among $2k$ numbers in $(Y(k+1)-3)$ comparisons. (A rigorous proof is given in [7].)

Let us consider an algorithm to find the least $(k+1)$ ones among $a_1, \ldots, a_{2k+2}$ in $Y(k+1)$ comparisons. First, several disjoint pairs will be compared:

$$a_{i(1)} \text{ and } a_{j(1)}, \ldots, a_{i(p)} \text{ and } a_{j(p)}\,.$$

Then a number $a_h$ in one of these pairs, say the lesser of the $i$-th pairs $a_{i(s)}$ and $a_{j(s)}$, will be compared with another number $a_{h'}$. After these comparisons, at most $(Y(k+1)-p-1)$ pairs will successively be chosen and compared.

Now consider the application of this algorithm to the selection of the least $k$ ones among

$$x_1, \ldots, x_{2k}\,. \tag{4}$$

We add to these numbers (4) two hypothetical elements, $x_{2k+1}$ and $x_{2k+2}$, which are not actually compared. The element $x_{2k+1}$ (or $x_{2k+2}$) is assumed to be less than (or greater than, respectively) any other numbers. Therefore, one of the least $(k+1)$ numbers among $x_1, \ldots, x_{2k+1}$ and $x_{2k+2}$ is $x_{2k+1}$ and the others are the least $k$ ones among (4). Therefore, applying the above mentioned algorithm, we can find the least $k$ numbers of (4) in $Y(k+1)$ comparisons.

Obviously, $Y(k+1)$ comparisons are not necessary: the comparisons involving $x_{2k+1}$ and/or $x_{2k+2}$ can be skipped. We shall show that at least three comparisons can be skipped when $k \geqslant 2$.

Let $\sigma$ be an arbitrary permutation in $S_{2k+2}$ satisfying the following conditions.

$$\sigma(i(s)) = 2k+1,$$
$$\sigma(j(s)) = 2k,$$

and

$$\sigma(h') \neq 2k+2.$$

If $a_{h'}$ is chosen as the lesser (or the greater) of the $t$-th pair, then we shall take:

$$\sigma(i(t)) = 2k-1,$$
$$\sigma(j(t)) = 2k-2.$$

Such a permutation $\sigma$ exists whenever $k \geqslant 2$. Since the indices $i(1), j(1), \ldots, i(p)$ and $j(p)$ are not important, we can start by comparing

$$x_{\sigma(i(1))} \text{ and } x_{\sigma(j(1))}, \ldots, x_{\sigma(i(p))} \text{ and } x_{\sigma(j(p))}\,.$$

The lesser of the $s$-th pair, $x_{2k+1}$ $(=x_{\sigma(i(s))})$, is then compared with $x_{\sigma(h')}$ $(\neq x_{2k+2})$.

Two comparisons can be skipped: the $s$-th and the $(p + 1)$-th. Since $x_{2k+2}$ must be compared elsewhere, one more comparison can be skipped.

This completes the proof of the Theorem 2.

TABLE I

| $k$ | Best upper bounds so far known | $Y(k)$ | $3k - 2$ | $y(k)$ |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 2 | 4 | 4 | 4 | 3 |
| 3 | 7 | 7 | 7 | 5 |
| 4 | 11 | ? | 10 | 7 |
| 5 | 15[a] | ? | 13 | 8 |

[a] K. Noshita [6].

Some results on $Y(k)$ for $k \leqslant 5$ are shown in Table I.

Exact evaluation of an operational entropy is often very difficult. In fact, there are many open problems on operational entropies. For instance, the values of $S(n)$ are known only when $n \leqslant 11$ and $n = 20, 21$ (see [3]). We do not yet know what $Y(4)$ is.

## Acknowledgments

The author would like to thank the people who participated in the discussions on operational entropies. The names of operational and informational entropies are due to Professor H. Takahasi, Director of Information Science Laboratory. Thanks are also due to Dr. R. L. Rivest of Stanford University who informed to the author recent results on the median computations.

## References

1. W. H. Burge, Sorting trees and measure of order, *Information and Control* **1** (1958), 181–197.
2. I. Flores, "Computer Sorting," Prentice-Hall, Englewood Cliffs, NJ, 1969.
3. L. R. Ford, Jr., and S. M. Johnson, A tournament problem, *Amer. Math. Monthly* **66** (1959), 387–389.
4. N. Yoneda, On certain ordering problems (Japanese), *Sugaku Seminar* **11** (1972), 34–35.
5. M. Blum, R. W. Floyd, V. Pratt, R. L. Rivest, and R. E. Tarjan, Linear time bounds for median computations, August 1971.
6. K. Noshita, On Yoneda's problem (private communication).
7. A. Nozaki, Operational entropies of grouping problems (unpublished memo, 1972).