HARVEST REPORT #10

Subject: Some Comments on the Problem of Sorting

By: Andrew Gleason

Date: December 18, 1956

December 18, 1956

HARVEST REPORT #10

Subject: Some Comments on the Problem of Sorting

We are given a collection of objects hereinafter called items which are linearly ordered in such a way that the relative order of any two items can be determined by direct comparison of the items themselves. Our objective is to arrange the items so that their linear sequence agrees with their intrinsic order. Many procedures have been described for bringing items into sequence and it is important to know which method is the fastest. The methods of information theory enable us to find certain theorems which are at least related to the problem of determining the fastest method of sorting.

In any machine computation, the steps can be divided roughly into two kinds, the administrative and the decisive. By the decisive steps we mean those places in the computation where the data itself enters in such a way as to affect the final result. The other steps are administrative; they include any sort of data rearrangement, counting routines (when the counting objectives are independent of the data), address modifications, print outs, etc. The distinction is not always clear-cut, but in the problem of sorting it is, or at least for the procedures which are widely used. Information theory proves that there are definite lower bounds to the number of decisive steps for any operation, so that one measure of efficiency in any procedure is obtained by comparing the actual number of decisive steps with the theoretical minimum. Unfortunately this measure takes no account of the administrative steps which can frequently be the principal part of the computation as far as overall time is concerned.

We shall discuss sorting by counting decisions, where we assume that all decisions are of the simple binary type: "Is item A ahead of item B ?"

Let us note that the fair estimate of the length of a process is the average length taken over all possible input data, weighted by the a priori probabilities of these data.

Theorem 1. Consider the problem of selecting one of n possible alternatives, all equally likely a priori, by a succession of binary decisions. The minimum, over all procedures, of the expected number of decisions is $\varphi(n)$ where:

$$\varphi(n) = p + 1 - \frac{2^p}{n} \qquad \text{if } 2^{p-1} \leq n \leq 2^p$$

(Proof will not be included here)

Theorem 2. If $C = 1 - \log_2 e + \log_2 \log_2 e \sim .086$, then
$$\log_2 n \leq \varphi(n) \leq \log_2 n + C.$$
The lower inequality is equality if n is a power of two. The upper bound is approximated when $n \sim 2^p/\log_2 e$.

In proving theorem 1, it is assumed that one may ask any sort of a question requiring a two-valued (yes or no) answer. The minimum will always be attained if at each stage the possible alternatives are divided into two groups as nearly equal as possible and asking whether the correct choice lies in the first group. In practice, we cannot always ask a question of this type by a simple step, so that $\varphi(n)$ can be well below the minimum of practical answers. Example: Given a file of n items, known to be sorted, and known to contain exactly one pair of duplicates, find them. In this problem there are n - 1 alternatives, since the pair can be the first and second card, the 2nd and 3rd, .., (n-1)st, nth. According to the theorem, the minimum length of a process for finding the pair is about $\log_2(n-1)$. This would be achieved if we could ask the question "Is the pair in the first half of the file?" But since there is no way of answering this question short of looking through the whole first half of the file, which itself would involve a lot of decisions, we cannot achieve the minimum of the theorem, or anything near it. It is clear that the best we can do using elementary questions, is to turn the file from the beginning comparing successive cards. Since we expect to find the pair halfway through on the average, the length of this process is 1/2 (n-1), a far cry from $\log_2(n-1)$ if n is large. Our process would be ideally efficient if we were asked to find the duplicates in a sorted file, without knowing that only one duplicate occurs. In that case we would have to go through the whole stack in any case so the process would have length n-1. But there are precisely 2^{n-1} ways that the file could contain duplicates. In the first problem, we seem to be obliged to reacquire information which we already knew, that most cards are unlike their neighbors.

Let us apply theorem 1 to the question of sorting a mixed file of n items. Our problem is equivalent to determining which of the n! rearrangements of the file actually puts it in sort, and these rearrangements are equally likely, since the file is presumed to be randomly mixed. Therefore the absolute lower bound to the number of binary decisions is $\varphi(n!)$. Now we cannot actually ask an arbitrary question; we are restricted to asking questions comparing just two items, and therefore this lower bound may not be attainable. By actual trials it was found that it is indeed attainable for n = 2, 3, 4, or 5, but there was no apparent system for doing so, and even if there were it would probably be unfeasible from the administrative view-point.

Consider the following method of sorting: Successively put each item into

the file in its correct place. Here we start with the first item, which can be filed in its place with no comparisons. Then the second item can be filed in one of two places, before or after the first and the mean number of decisions in $\varphi(2)$. The third item goes in one of three places and the most efficient procedure will require $\varphi(3)$ decisions on the average. In general, the k-th card can be added in k ways and the correct decision will require $\varphi(k)$ steps on the average. This value can actually be achieved by comparing the new item with the middle item of the file, then with the item in the middle of the appropriate half, etc. The whole process will require $\varphi(1) + \varphi(2) + \text{-----} + \varphi(n)$ steps on the average. Using the estimate of theorem 2 we have

$$\log_2 n! = \log_2 1 + \log_2 2 + \text{-----} + \log_2(n) \leq \varphi(1) + \text{---} + \varphi(n) \leq$$

$$\leq (\log_2 1 + .086) + (\log_2 2 + .086) + \text{----} + (\log_2 n + .086)$$

$$= \log_2 n! + .086n.$$

We can show that on the average

$$\varphi(n) - \log_2 n \sim .056$$

so that

$$\varphi(1) + \text{-----} + \varphi(n) \sim \log_2 n! + .056n$$

This proves that the method of successive filing of single items is very close to the absolute lower bound percentage-wise. Thus for $n = 1,000,000$ we find that $\varphi(1,000,000!) \sim 18.5\,(10^6)$ so that $\varphi(1) + \text{-----} + \varphi(1,000,000)$ is only 0.3% larger.

We consider the method of merging into blocks of two, then four, etc. We assume that the block sizes are always exact powers of two, and, for convenience, that the total number of items is a power of 2, say 2^l. We use the ordinary process for merging two blocks of size 2^k. This will never take more than $2^{k+1} - 1$ comparisons, and on the average it will take $2^{k+1} - 2 + \frac{2}{2^k + 1}$. Then the entire process will take on the average

$$2^{p-1}\left(2 - 2 + \frac{2}{1+1}\right) + 2^{p-2}\left(2^2 - 2 + \frac{2}{2+1}\right) + 2^{p-3}\left(2^3 - 2 + \frac{2}{4+1}\right) + \text{----}$$
$$+ 2^0\left(2^p - 2 + \frac{2}{2^{p-1}+1}\right)$$

steps. The principal terms work out to be

$$2^p\,(p - 2 + \theta)$$

where $\theta = \frac{1}{1(1+1)} + \frac{1}{2(2+1)} + \frac{1}{4(4+1)} + \quad + \frac{1}{2^{p-1}(2^{p-1}+1)} \sim .736$

Using Sterling's approximation to log n!, we find

$$\varphi(2^p!) \sim (p - \log_2 e)\, 2^p$$

Therefore, this method is proportionately longer than perfect by

$$\frac{\log_2 e - 2 + \theta}{p - \log_2 e} = \frac{1.442 - 2 + .736}{p - \log_2 e} = \frac{.178}{p - 1.442}$$

For p = 20, i.e. 2^{20} or about 1,000,000 items this is only 1%.

When the method of successive collation is applied to blocks of mixed sizes, it is necessary to continually check for the end of the block. This has the effect of nearly doubling the number of comparisons, while its effect on the number of passes through is to reduce it by one, hence this method will always be markedly inefficient when judged by the simple criterion of number of comparisons made. This is not to say that the method is unsatisfactory. Rather it suggests that the number of comparisons is not an appropriate measure of efficiency.

The two procedures discussed in detail also show that the measure of efficiency is inappropriate. Single item filing can be done very efficiently in terms of number of comparisons, but in practice it is not easy to do on tapes. It would require a great deal of administrative time. The factor of two in comparisons between the fixed-sized and random-sized block methods is also illusory, because in the fixed size method a count and corresponding interior comparison must be made to identify the ends of blocks. Unless the method of determining precedence between items is complicated, the random sized block method will be easier. In this method, hard comparisons for ends of blocks can be avoided if an extra control symbol is added to the beginnings of blocks. This reduces the comparison to a simple search for the control symbol which is almost always quicker. This gives up any random increases in block size, but these are in any event rare after the first pass.

Theorem: Suppose we are given n equally likely hypothesis concerning an object and wish to determine which is valid by binary conbinational questioning (i.e. each question we ask receives a yes-or-no answer). Each questioning

procedure yields an expected number of questions and also a maximum number of questions. Among all questioning procedures, the minimum value of the expected number of questions is

$$E(n) = (p+1) - \frac{2^p}{n} \qquad \text{where } p \text{ is the least integer for which } n \leqslant 2^p.$$

Among all questioning procedures, the minimum value of the maximum number of questions is

$$M(n) = p$$

Both of these bounds are attained by the procedure of always dividing the hypotheses remaining into two equal groups (or groups as nearly equal as possible, i. e. within one) and asking in which group does the correct hypothesis lie.

Proof: Whatever question we may ask, the two answers have the effect of dividing the hypotheses into two disjoint groups, so we may as well consider every question to be of the form "Is it in this group?". Now since the hypotheses are assumed to be a priori indistinguishable, the result of asking the questions question "Is it in this group of k?" is either 1) to put us in the position of determining which of k hypotheses is valid, which happens if the answer is "yes", an event of probability k/n or 2) to put us in the position of determining which of n-k hypotheses is valid, which happens if the answer is "no", an event of probability (n-k)/n. In any case, we then proceed to ask about the reduced group by whatever procedure is optimum for the size in question. If, then, E(n) is the expected number of questions by the optimum procedure for a group of n, the expected number of questions by the procedure outlined above is

$$1 + \frac{k}{n} E(k) + \frac{n-k}{n} E(n-k).$$

The optimum procedure for n-hypotheses is, then obtained by choosing the minimum value of this expression for all possible values of k (i. e. 1, 2, ..., n-1). Thus

$$E(n) = 1 + \min_k \left\{ \frac{k}{n} E(k) + \frac{n-k}{n} E(n-k) \right\} \qquad *$$

This gives us a recurrence from which E can easily be calculated. We may however achieve a closed form by first putting

$$F(n) = nE(n)$$

$$\text{Then } F(n) = n + \min_k \left\{ F(k) + F(n-k) \right\}, \qquad **$$

Suppose that $n = 2^{p-1} + q$ where $0 \leqslant q < 2^{p-1}$

Put $F_o(n) = (p - 1)\, 2^{p-1} + (p + 1)\, q$

First, we note that $F_o(n) = n + F_o\left(\left\lceil \frac{n}{2} \right\rceil\right) + F_o\left(n - \left\lceil \frac{n}{2} \right\rceil\right)$

by direct computation in two cases according as n is even n odd. Second, observing that F_o in a linear interpolation of the function $x \log_2 x$, agreeing with the latter for powers of two, it is clear that F_o is convex, so that

$$\min_{R=1,\ldots n-1} \left\{ F_o(k) + F_o(n-k) \right\} = F_o\left(\left\lceil \frac{n}{2} \right\rceil\right) + F_o\left(n - \left\lceil \frac{n}{2} \right\rceil\right).$$

Therefor F_o satisfies the recursion **. Hence $F = F_o$.

Writing $F(n) = p\, 2^{p-1} + pq + q - 2^{p-1} = p \cdot n + n - 2^p$

We get $E(n) = p + 1 - \frac{2^p}{n}$ as stated.

Secondly, the choice $k = \left\lceil \frac{n}{2} \right\rceil$ is always a suitable choice of k in *, so that the minimum value of E is in fact attained with a process which always divides the stacks evenly as nearly as possible. Finally, if $n \leq 2^p$, then at the q-th stage all groups will be less than n equal to 2^{p-n} in number, and after p stages, no group will have more than 1 member, i. e. the problem is done. Evidently $M(*) \geq 2^p$.

Theorem: $0 \leq E(n) - \log_2 n \leq 1 - \log_2 e + \log_2 \log_2 e \sim .086$

Proof: $F(n)$ is linear between $n = 2^{p-1}$ and $n = 2^p$ and agrees with $n \log_2 n$ at these two points. The latter being strictly convex
$n \log_2 n \leq F(n)$ for all n.
Hence $\log_2(n) \leq E(n)$ for all n.
In the interval

$2^{p-1} \leq n \leq 2^p$ we may write

$$E(n) = p + 1 - \frac{2^p}{n}$$

$$E(n) - \log_2 n = (p + 1) - \frac{2^p}{n} - \log_2 n$$

The maximum value of the function

$$p + 1 - \frac{2^p}{x} - \log_2 x$$

is achieved when

$$\frac{2^p}{x^2} - \frac{\log_2 e}{x} = 0 \quad \text{or}$$

$$x = \frac{2^p}{\log_2 e}$$

when $p + 1 - \frac{2^p}{x} - \log_2 x = (p + 1) - \log_2 e - p + \log_2 (\log_2 e)$

$$= 1 - \log_2 e + \log_2 \log_2 e.$$

$\therefore$ $E(n) - \log_2 n \leq 1 - \log_2 e + \log_2 \log_2 e$ for all n.

Comments: Evidently not more than 2^q hypotheses can be distinguished with q questions, and exactly this many can be done with q questions. If the number of hypotheses is not a power of 2, then one cannot make all questions efficient, so there is bound to be a loss due to "breakage". The discrepancy

$$E(n) - \log_2 n$$

measures this loss. It is remarkably small.

Application to sorting problem:

1. To put the n-th number in sequence in a file already containing n items properly sequenced. There are n possible places to put the new item. Therefore the minimum expected number of comparisons is at least E (n). But since it is possible to always make the comparison so as to split the possibilities as nearly evenly as possible, i. e. always compare the new item with the middle item of the string in which it belongs, the number of comparisons need not be more than E (n).

2. To arrange n items in sequence, with no a priori information about their order. There are n! rearrangements possible. To sequence the file correctly we must determine (possibly not explicitly) which of these arrangements to correct. Hence any procedure must have an expected number of comparisons $\geq E(n!)$. It is by no means clear that the questions which achieve this expectation can actually be phrased in the form of simple comparisons between two items, so it may not be possible to achieve this bound. However, we can see that the method of adding one more item to the file after another has an expectation

$$E(1) + E(2) + \cdots + E(n)$$

$$= \log 1 + \log 2 + \cdots + \log n + \sum (E(i) - \log i) \leq \log_2 n! + n(.086)$$

AG/jh A. Gleason