JOURNAL OF COMPUTER AND SYSTEM SCIENCES 24, 162–166 (1982)

Finding the Median Distributively

MICHAEL RODEH

IBM Israel Scientific Center, Haifa, Israel

Received October 25, 1979; October 15, 1981

We consider the problem of computing the median of a bag of $2n$ numbers by using communicating processes, each having some of the numbers in its local memory. The memories are assumed to be disjoint. For two processes an algorithm is given. Its time and space complexity is linear while the communication complexity is $2 \log_2 n$. A lower bound of $\log_2 n$ on the communication complexity is derived. Thus the algorithm is optimal up to a constant.

1. INTRODUCTION

Finding the median of a bag (a set which may have repeated elements) of numbers is a problem that has attracted many researchers. Blum *et al.* [1] developed the first linear algorithm to find the median. Schönhage *et al.* [6] developed another algorithm which performs fewer comparisons on the worst case. In both algorithms the given numbers are assumed to reside in the memory of a single processor. When having more numbers than the size of the memory one can either look for an algorithm which uses auxiliary memory or distribute the numbers among several communicating processors. In the sequel we concentrate on the problem of designing an efficient distributive algorithm to compute the median of a given bag of numbers. The simplest case is when two identical communicating processors are available and each one of them has half of the numbers in its possession.

Let U be a given bag of numbers and let $|U|$ denote its cardinality. Given a bag U, the non-decreasing ordering of the elements of U is denoted by $\mathbf{U}$. If $|U| = 2n$ for some integer n then *median* $1(U)$ and *median* $2(U)$ are the nth and $n+1$st elements of $\mathbf{U}$. If $|U| = 2n+1$ then the median of U is denoted by both *median* $1(U)$ and *median* $2(U)$. Notice that *median* $1(U)$ and *median* $2(U)$ are elements of U.

A median of a bag U induces a partition of the elements of U into two sub-bags. If the elements of U are distinct then the two sub-bags have about the same size. If there are repeated elements then some additional mechanism may be employed in order to obtain a partitioning of U into sub-bags of about equal size. Consider the case of having a bag U partitioned into two sub-bags A and B such that $|A| = |B| = n$ and assume that A is stored in the memory of one processor and B is stored in the memory of the other. In addition, assume that the memories are disjoint (no sharing

0022-0000/82/020162-05$02.00/0

of memory is allowed). An importat property of the median is that it induces a partition of the given bag.

It could be advantageous to preserve this property also in the distributive case. Assume that the *distributive median* μ of A and B has been defined somehow, and assume for a moment that all the elements of U are distinct. Then it is required that μ will induce a partitioning of A and B into $A1$, $A2$ and $B1$, $B2$, respectively, such that

$$|A1| + |B1| = |A2| + |B2|.$$

Using

$$|A1| + |A2| = |B1| + |B2|$$

we get

$$|A1| = |B2| \qquad \text{and} \qquad |B1| = |A2|.$$

For every $a1 \in A1$, $a2 \in A2$, $b1 \in B1$, $b2 \in B2$, the relationships between $A1$, $A2$, $B1$, $B2$ may be expressed by:

$$a1, b1 \leqslant a2, b2.$$

Choosing the distributive median μ to be an element of U leads to a nonsymmetric definition since μ may belong either to A or to B but in general will not belong to both. Noticing that $|U|$ is even one could consider using *median*$1(U)$ and *median*$2(U)$ as distributive medians. Still it may happen that both of them belong to A (or B) and no symmetry is gained.

The above discussion leads to the following definition: A *distributive median* of the pair (A, B), where $|A| = |B| = n$, is a number m such that:

(i) $0 \leqslant m \leqslant n$.

(ii) Let $A1$ contain the m smallest elements of A, $A2$ the $n-m$ largest elements of A, $B1$ the $n-m$ smallest elements of B and $B2$ the m largest elements of B. Then for every $a1 \in A1$, $a2 \in A2$, $b1 \in B1$ and $b2 \in B2$,

$$a1, b1 \leqslant a2, b2.$$

Notice that A and B may have more than one distributive median. For $U = A \cup B$, where $A = (3, 1, 3, 10)$ and $B = (8, 4, 2, 3)$, the median is either 2 or 3. For $m = 3$ we have $A1 = (1, 3, 3)$, $A2 = (10)$, $B1 = (2)$, $B2 = (3, 4, 8)$.

The definition of the distributive median is symmetric and it directly induces a partition of A and B (into $A1$, $A2$ and $B1$, $B2$ respectively). It further has the advantage that once the distributive median is known to both processes, each one of them is able to perform the partition by itself without consulting the other process.

2. A Distributive Algorithm for Computing the Distributive Median

The proposed algorithm consists of two communicating processes, P_A and P_B having A and B, respectively, in their local memory. In every step, P_A communicates with P_B and as a result of the information gained by this communication P_A decides that certain elements of A necessarily belong to $A1$ or to $A2$ (which are initially considered empty), thereby reducing the cardinality of A and increasing either that of $A1$ or that of $A2$. When all the elements of A are exhausted P_A terminates. The structure of P_B is similar.

To be more specific, P_A receives by communication an element of B from P_B. Knowing that this element is $\mathit{median}2(B)$ it compares it to $\mathit{median}1(A)$. If $\mathit{median}1(A) < \mathit{median}2(B)$ then the $\mathit{ceiling}(n/2)$ (hereafter denoted by $\lceil n/2 \rceil$) smallest elements of A may be added to $A1$. These elements are not larger than at least n elements of $A \cup B$ (the union here preserves repetitions): the largest $\mathit{floor}(n/2)$ (hereafter denoted by $\lfloor n/2 \rfloor$) elements of A and the largest $\lceil n/2 \rceil$ elements of B. If $\mathit{median}1(A) > \mathit{median}2(B)$ then the largest $1 + \lfloor n/2 \rfloor$ elements of A may be added to $A2$ (by an argument similar to that above). The case of equality is even simpler. In any of the above possibilities, the variable m (initially 0) keeps track of the cardinality of $A1$. P_B is dual to P_A and is very similar to it ($\mathit{median}2(B)$ replaces $\mathit{median}1(A)$ etc.).

3. The Complexity of Finding the Distributive Median

The algorithm utilizes three types of resources: time, space and communications. The communication complexity must be considered separately since communication cost is in general incomparable to time or space.

3.1. *Time and Space Complexity*

Let A^i and B^i be the bags from which P_A and P_B choose medians during the ith phase. First observe that updating A^i and B^i in order to obtain A^{i+1} and B^{i+1} requires partitioning rather than sorting. Using one of the linear algorithms to compute the median [1, 6], the ith iteration of P_A and P_B takes $c \cdot |A^{i-1}|$ time for some constant c independent of i. Since $|A^{i-1}| \geqslant 2\,|A^i|$ the total time required by P_A (and P_B) is $O(|A|)$ and therefore the algorithm is linear in the size of A. It must be noticed that the time required by P_A is about equal to the time required to find the median of A and B by a nondistributive algorithm. Therefore no saving in the finishing time (the time it takes for all processes to finish their computations) is observed although P_A and P_B operate concurrently. The space requirements of the algorithm are linear as well.

3.2. *Communication Complexity*

In each iteration P_A exchanges one element with P_B. As a result the size of the problem is reduced by at least a factor of 2. Therefore the number of exchanges is bounded by $\log_2 n$. If only sending elements is counted then there are $2 \log_2 n$ communications.

Below we prove that $\log_2 n$ is a lower bound on the number of communications. To this end consider the following special case in which P_B knows the distributive median in advance but P_A does not. Assume that A consists only of elements equal to 1, and that B contains m elements equal to 2 and $n - m$ elements equal to 0. The distributive median is obviously m. The only remaining problem is to let P_A know that the answer is m. The value of m may be any number in the range 0 to n. Therefore, at least $\log_2 n$ characters (0 or 2) must be sent, leading to a $\log_2 n$ lower bound on the communication complexity.

If P_B knows the value m but there are many distinct elements in B then it can first send some of its elements to P_A and then use them as an alphabet to transmit m. The number of communications is then reduced to at most $2 \cdot l$, where l is a solution of $n = l^l$. Notice that $l = o(\log_2 n)$.

This proof uses the constraints which we have imposed on the communication, namely, that only elements of A may be sent to P_B and similarly for P_B. However, as pointed out by Pippenger [4] a somewhat stronger result can be proven: Any algorithm for determining the median or the distributed median of $2n$ elements, which are initially partitioned into two bags A and B of equal sizes, must perform at least $\log_2 n$ comparisons between disjoint pairs of elements (each pair containing one element from A and one from B) even if comparisons between two elements from A or two elements from B are allowed for free. This can be shown by using an adversary argument of the type used by Pratt and Yao [5]: Suppose that an element $a \in A$ is compared to the elements of B. If $a < \mathrm{A}_{\lceil n/2 \rceil}$ then the adversary tells us that a is smaller than all the elements of B; otherwise it is bigger than all elements of B. Assume that such a comparison is charged as one comparison only. Then at most half of the elements of A may be determined as not being the median, but still there is an uncertainty about the other half. Moreover, nothing is known about the relative sizes of this other half and the elements of B. In this way, one comparison may reduce the size of the problem by at most a half, thus proving the desired result. Applying this observation to the distributive case shows that at least $\log_2 n$ communications are required independent of whether there are repeated elements or not.

Another question related to the problem of communication complexity is what happens if more than one element at a time may be exchanged between P_A and P_B. Then, as can easily be verified, as a result of one exchange of k numbers the size of A and B may be reduced by a factor of $k + 1$. Therefore, the total number of exchanges becomes $\log_{k+1} n$ rather than $\log_2 n$.

The algorithm given above may be implemented on a single processor. Then it may be considered as a recursive algorithm. As such it has the following formula characterizing the time complexity:

$$f(2n) = 3f(n) + 2n,$$

where $3f(n)$ is the sum of the times required to find the medians of A and B, and the time required to find the distributive median of the reduced A and B. Partitioning A and B takes $2n$ units of time. The solution to this formula is $O(n^{\log_2 3})$. We see that the algorithm is not linear when considered as a nondistributive algorithm and therefore is not optimal. Notice that the cost of the partitioning operations is only linear and thus could be discarded without introducing any significant improvement.

4. Conclusion

As discussed above, the definition of the distributive median does not follow directly from the definition of the standard median. The algorithm for solving it is optimal up to a constant factor. However, its generalization to an unbounded number of processes is not linear, and therefore it is not clear whether it is optimal. An interesting question is how to design the communication links between processes so that the communication complexity is reduced. A simple approach is to use the divide and conquer technique which uses a communication graph with a tree structure. In general, it seems that other communication graphs may leads to better algorithms for certain problems.

Acknowledgments

I would like to thank N. Francez, S. Katz and Y. Shiloach for helpful discussions.

References

1. M. Blum, R. Floyd, V. Pratt, R. Rivest, and R. Tarjan, Time bounds for selection, *J. Comput. System Sci.* 7 (1973), 448–461.
2. C. A. R. Hoare, Communicating sequential processes, *Comm. ACM* **21** (1978), 666–677.
4. N. Pippenger, Private communication.
5. V. Pratt and F. Yao, On lower bounds for computing the *i*th largest element, *in* "Proceedings, 14th Ann. IEEE Symp. on Switching and Automata Theory (1973), pp. 70–81.
6. A. Schönhage, M. Paterson, and N. Pippenger, Finding the median, *J. Comput. System Sci.* **13** (1976), 184–199.