MULTIPLE QUICKSELECT —
HOARE'S FIND ALGORITHM FOR SEVERAL ELEMENTS

HELMUT PRODINGER

Department of Algebra and Discrete Mathematics
Technical University of Vienna, Austria

ABSTRACT. Hoare's Find algorithm can be used to select p specified order statistics $j_1, j_2, \dots, j_p$ from a file of n elements simultaneously. We give precise formulæ for both the average number of passes and the average number of comparisons. Averaging again over all possible subsets of p elements, we get results of Lent and Mahmoud as corollaries.

— April 26, 1996 —

1. INTRODUCTION

Hoare's FIND algorithm [3] uses the idea of *Quicksort* [5] in order to select the jth element of a file of n elements. In each partitioning step, a certain element will be brought into its correct position k. In the process, elements are moved—those brought to the left of it are smaller, and those brought to the right are larger. If $k = j$, the element is found; if $k < j$, one has to continue on the right side, searching for the $(j-k)$th element; if $k > j$, one has to continue on the left side, still searching for the jth element.

Two parameters are of interest: the (average) number of passes (recursive calls) $P[n; j]$ and the (average) number of comparisons $C[n; j]$. In Knuth's book [5] we already find the values

$$P[n; j] = H_j + H_{n+1-j} - 1$$

and

$$C[n; j] = 2\Big(n + 3 + (n+1)H_n - (j+2)H_j - (n+3-j)H_{n+1-j}\Big)$$

where $H_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n}$ denotes the nth harmonic number; compare also [6] and [8].

In [4], this analysis was extended to the case of median–of–three partitioning.

Hoare's idea can be extended as follows: Assume that we simultaneously search for the p elements with ranks $j_1, j_2, \dots, j_p$ where $1 \le j_1 < j_2 < \cdots < j_p \le n$ is a fixed set of p values. Then, according into which interval of the values $j_1, \dots, j_p$ the pivot element falls, we may have found one of the elements and/or have to continue on both sides, but with

File: lint3.tex

smaller sets. The idea is quite simple; a nice description of the algorithm can be found in [7].

Multiple Quickselect is also easy to code and behaves very well in all cases in practice—as the theoretical analysis predicts.

In the following we will give explicit formulæ for both the average number of passes $P[n;j_1,\dots,j_p]$ and the average number of comparisons, $C[n;j_1,\dots,j_p]$. Also of interest are the *grand* averages,

$$\mathcal{P}_{n,p} = \frac{1}{\binom{n}{p}} \sum_{1\le j_1<j_2<\dots<j_p\le n} P[n;j_1,\dots,j_p]$$

and

$$\mathcal{C}_{n,p} = \frac{1}{\binom{n}{p}} \sum_{1\le j_1<j_2<\dots<j_p\le n} C[n;j_1,\dots,j_p]\ .$$

They were already considered in [7], and our explicit formulæ extend the asymptotic formulæ therein.

2. The average number of passes

We will give the recursion in the case of a two element set $\{j<l\}$. The general case is a straightforward extension which we omit as the formula is quite involved;

$$\begin{aligned} P[n;j,l] = 1 &+ \frac{1}{n}\sum_{1\le k<j} P[n-k;j-k,l-k] \\ &+ \frac{1}{n}\sum_{j<k\le l} P[k-1;j] + \frac{1}{n}\sum_{j\le k<l} P[n-k;l-k] \\ &+ \frac{1}{n}\sum_{l<k\le n} P[k-1;j,l]\ . \end{aligned}$$

The extra 1 counts one pass of the algorithm, and the factor $\frac{1}{n}$ is the probability that the kth element was chosen as a pivot element. If k is smaller than j or larger than l, we still search for two elements on one side, otherwise we search for one element on each side (unless we already found one of them; the case $k=j$ of $k=l$).

Theorem 1.

$$P[n;j_1,\dots,j_p] = H_{j_1} + H_{n+1-j_p} + 2\sum_{t=2}^{p} H_{j_t+1-j_{t-1}} - 2p + 1\ . \quad \diamondsuit$$

We don't give a full proof of this theorem, because it is long and completely routine. The proof just shows that the given explicit formula fulfills the recursion, so that the result is proved by induction. Not more is required for that than the formula

$$\sum_{1\le k<n} H_k = n\big(H_n - 1\big)\ .$$

It is however not all that easy to *find* the formula. The formula was guessed and verified through extensive computer experiments with the computer algebra system MAPLE.

Another line of proving the theorem, as suggested to us by C. Martínez, would be to work with generating functions. For instance

$$\sum_{1\le j\le n} P[n;j]u^j z^n = \frac{1}{1-uz}\log\frac{1}{1-uz} + \frac{u}{(1-uz)(1-z)}\log\frac{1}{1-z} - \frac{uz}{(1-z)(1-uz)} .$$

And it is equally sufficient to show that this function fulfills the differential equation which is the equivalent in terms of generating functions to the recursion for the $P[n;j]$'s. In the general case, variables $u_1,\dots,u_p$ and z must be used.

It might be instructive to give the special case of two elements explicitly:

$$P[n;j,l] = H_j + 2H_{l+1-j} + H_{n+1-l} - 3 .$$

Observe that this formula (as well as the general one) fulfills the identity $P[n;j,l] = P[n;n+1-l,n+1-j]$, which is obvious from the symmetry of the algorithm.

To get some feeling about the average number of passes, let us assume that $j_t \sim a_t \cdot n$, with some constants $0 < a_1 < \cdots < a_p < 1$, where p is fixed. Then

$$P[n;j_1,\dots,j_p] = 2p\log n + \mathcal{O}(1) .$$

It is also worth noticing that the formula for $P[n;j_1,\dots,j_p]$ remains correct if we relax the condition $1 \le j_1 < j_2 < \cdots < j_p \le n$ to $1 \le j_1 \le j_2 \le \cdots \le j_p \le n$.

If $p = n$, this just means *sorting*, and our formula tells us

$$P[n;1,2,\dots,n] = n,$$

which is well known (and obvious!); it comes out for instance from the recursion

$$P_n = 1 + \frac{1}{n}\sum_{k=1}^{n}\Big(P_{k-1} + P_{n-k}\Big) .$$

3. THE AVERAGE NUMBER OF COMPARISONS

The recursion for the number of comparisons is almost the same as before, except that the additional 1 (counting one pass) is now replaced by $n-1$ which is the number of comparisons required to bring the kth element into its correct position. Let us again consider the case of a two element set $\{j < l\}$,

$$\begin{aligned} C[n;j,l] = n - 1 &+ \frac{1}{n}\sum_{1\le k<j} C[n-k;j-k,l-k] \\ &+ \frac{1}{n}\sum_{j<k\le l} C[k-1;j] + \frac{1}{n}\sum_{j\le k<l} C[n-k;l-k] \\ &+ \frac{1}{n}\sum_{l<k\le n} C[k-1;j,l] . \end{aligned}$$

Observe that we only count the comparisons which are necessary to bring the pivot element into its correct position k. Extra comparisons of k with the numbers $j_1,\dots,j_p$ are not considered. (This would give an additional contribution of roughly $\log p$ comparisons.)

Theorem 2.

$$C[n;j_1,\dots,j_p] = 2n + j_p - j_1 + 2(n+1)H_n - 2(j_1+2)H_{j_1} - 2(n+3-j_p)H_{n+1-j_p}$$
$$- 2\sum_{t=2}^{p}(j_t + 4 - j_{t-1})H_{j_t+1-j_{t-1}} + 8p - 2\ . \quad \diamondsuit$$

In this instance it is even more true that the *finding* of the formula is the nontrivial and the proof by induction is the routine part (which we therefore skip). It requires additionally the formula

$$\sum_{1\le k<n} k\,H_k = \binom{n}{2}\left(H_n - \tfrac{1}{2}\right)\ .$$

The formula in Theorem 2 fulfills the required symmetry $C[n;j_1,\dots,j_p] = C[n;n+1-j_p,\dots,n+1-j_1]$ and remains valid if some of the j's are equal.

Again, if we let $j_t \sim a_t \cdot n$, we get

$$C[n;j_1,\dots,j_p] = n\cdot\Big[2 + a_p - a_1 - 2a_1\log a_1 - 2(1-a_p)\log(1-a_p)$$
$$- 2\sum_{t=2}^{p}(a_t - a_{t-1})\log(a_t - a_{t-1})\Big] + \mathcal{O}(\log n)\ .$$

It is interesting to investigate when the factor of n gets maximal. For instance, in the case of two elements we see

$$a_1 = \frac{1}{2+\sqrt{e}}\,, \qquad a_2 = 1 - a_1\,;$$

for three elements we find

$$a_1 = a_3 = \frac{1}{2}\cdot\frac{1+2\sqrt{e}}{1+\sqrt{e}}\,, \qquad a_2 = \frac{1}{2}\,,$$

etc. If $p = n$, this just means *sorting*, and our formula tells us

$$C[n;1,2,\dots,n] = 2(n+1)H_n - 4n\,,$$

which is well known; it comes out for instance from the recursion

$$C_n = n - 1 + \frac{1}{n}\sum_{k=1}^{n}\Big(C_{k-1} + C_{n-k}\Big)\ .$$

Remark. Our explicit formulæ translate directly into the costs of the computation of several classes of (order–)statistics, such as e.g. the Hodges–Lehmann statistics. We merely give this general remark and refer to [7] and [9].

4. The grand average of passes

As we have already announced, we compute

$$\sum_{1\le j_1<j_2<\dots<j_p\le n} P[n;j_1,\dots,j_p]\ .$$

To achieve this goal, we first compute

$$\begin{aligned}\sum_{1\le k\le n} H_k\binom{n-k}{p-1} &= [z^n]\frac{1}{1-z}\log\frac{1}{1-z}\cdot\frac{z^{p-1}}{(1-z)^p}\\ &= [z^{n+1-p}]\frac{1}{(1-z)^{p+1}}\log\frac{1}{1-z}\\ &= \binom{n+1}{p}(H_{n+1}-H_p)\ .\end{aligned}$$

Here, we used some generating functions that are discussed in great detail in [1] and [2]. Then we see

$$\begin{aligned}\sum_{1\le j_1<\dots<j_p\le n} H_{j_1} &= \sum_{1\le j\le n} H_j\binom{n-j}{p-1}\\ &= \binom{n+1}{p}(H_{n+1}-H_p) = \sum_{1\le j_1<\dots<j_p\le n} H_{n+1-j_p}\end{aligned}$$

and

$$\begin{aligned}\sum_{t=2}^{p}\sum_{1\le j_1<\dots<j_p\le n} H_{j_t+1-j_{t-1}} &= \sum_{1\le j<l\le n} H_{l+1-j}\binom{n-(l+1-j)}{p-2}\\ &= \sum_{k=2}^{n} H_k(n+1-k)\binom{n-k}{p-2}\\ &= (p-1)\sum_{k=2}^{n} H_k\binom{n+1-k}{p-1}\\ &= (p-1)\sum_{k=1}^{n+1} H_k\binom{n+1-k}{p-1} - (p-1)\binom{n}{p-1}\\ &= (p-1)\binom{n+2}{p}(H_{n+2}-H_p) - (p-1)\binom{n}{p-1}\ .\end{aligned}$$

Collecting all contributions and dividing by $\binom{n}{p}$, we get

Theorem 3.

$$\mathcal{P}_{n,p} = \frac{2p(n+1)^2}{(n+2-p)(n+1-p)}\Big(H_{n+1} - H_p\Big) + 1 - 2p - \frac{2(p-1)^2}{n+2-p} \ . \quad \diamondsuit$$

The classical case is $p = 1$, and then

$$\mathcal{P}_{n,1} = 2\Big(1 + \frac{1}{n}\Big)H_n - 3 \ .$$

Computing an asymptotic equivalent, we get

Corollary 1. *For p fixed and $n \to \infty$ we have*

$$\begin{aligned} \mathcal{P}_{n,p} = {} & 2pH_n - 2pH_p - 2p + 1 + 2p(2p-1)\frac{H_n}{n} \\ & - \frac{2p(2p-1)H_p + 2(p^2 - 3p + 1)}{n} + \mathcal{O}\Big(\frac{\log n}{n^2}\Big) \ . \end{aligned}$$

5. The grand average of comparisons

For this, we need the following auxiliary results:

Sum 1.

$$\begin{aligned} \sum_{1 \le j_1 < \cdots < j_p \le n} (j_p - j_1) &= \sum_{1 \le j < l \le n} (l - j)\binom{l-1-j}{p-2} \\ &= (p-1) \sum_{1 \le j < l \le n} \binom{l-j}{p-1} \\ &= (p-1) \sum_{1 \le k \le n} \binom{k}{p-1}(n-k) \\ &= (p-1)\binom{n+1}{p+1} \ . \end{aligned}$$

Sum 2.

$$
\begin{aligned}
\sum_{1\le j_1<\cdots<j_p\le n}(j_1+2)H_{j_1} &= \sum_{j=1}^{n}(j+2)H_j\binom{n-j}{p-1}\\
&= \sum_{j=1}^{n}\big[(n+3)-(n+1-j)\big]H_j\binom{n-j}{p-1}\\
&= (n+3)\sum_{j=1}^{n}H_j\binom{n-j}{p-1} - p\sum_{j=1}^{n+1}H_j\binom{n+1-j}{p}\\
&= (n+3)\binom{n+1}{p}\left(H_{n+1}-H_p\right) - p\binom{n+2}{p+1}\left(H_{n+2}-H_{p+1}\right)\\
&= \sum_{1\le j_1<\cdots<j_p\le n}(n+3-j_p)H_{j_p}\ .
\end{aligned}
$$

Sum 3.

$$
\begin{aligned}
&\sum_{t=2}^{p}\sum_{1\le j_1<\cdots<j_p\le n}(j_t+4-j_{t-1})H_{j_t+1-j_{t-1}}\\
&\qquad= \sum_{1\le j<l\le n}(l+4-j)H_{l+1-j}\binom{n-(l+1-j)}{p-2}\\
&\qquad= (p-1)(n+5)\binom{n+2}{p}\left(H_{n+2}-H_p\right)\\
&\qquad- (p-1)p\binom{n+3}{p+1}\left(H_{n+3}-H_{p+1}\right) - 4(p-1)\binom{n}{p-1}\ .
\end{aligned}
$$

Now we collect all contributions and divide by $\binom{n}{p}$ and, after several simplifications, we obtain the following *explicit formula*.

Theorem 4.

$$
\begin{aligned}
\mathcal{C}_{n,p} = \frac{1}{(n+2-p)(n+1-p)}\Big[&(2H_p+1)n^3 - 8pH_nn^2 + 4((p+2)H_p+p)n^2\\
&+ 2p(p-9)H_nn + (-5p^2+p-1+2(4p+5)H_p)n\\
&+ 2p(p-5)H_n + 4(p+1)H_p - p(p+7)\Big]\ . \quad \diamondsuit
\end{aligned}
$$

The classical case is $p=1$, and then

$$
\mathcal{C}_{n,1} = 3n - 8H_n + 13 - \frac{8H_n}{n}\ .
$$

Computing an asymptotic equivalent, we get

Corollary 2. *For p fixed and $n \to \infty$ we have*

$$\mathcal{C}_{n,p} = (2H_p+1)n - 8pH_n + 2(4p+1)H_p + 3(2p-1) - 2p(7p-3)\frac{H_n}{n} + \frac{2p(7p-3)H_p + 2(p-3)(3p-1)}{n} + \mathcal{O}\Big(\frac{\log n}{n^2}\Big) \ .$$

Acknowledgment. I thank Prof. H. Mahmoud for sending me his preprint and clarifying some of the ideas therein as well as Prof. C. Martínez for interesting suggestions. Comments of two referees helped to improve the presentation of the paper.

References

1. R. Graham, D.E. Knuth, and O. Patashnik, *Concrete Mathematics*, Addison-Wesley, 1989.
2. D. Greene, D.E. Knuth, *Mathematics for the Analysis of Algorithms*, Birkhäuser, 1981.
3. C.A.R. Hoare, *Find (Algorithm 65)*, Comm. of the ACM **4** (1961), 321–222.
4. P. Kirschenhofer, C. Martínez, and H. Prodinger, *Analysis of Hoare's Find algorithm with median–of–three partition*, submitted (1995).
5. D.E. Knuth, *The Art of Computer Programming Vol. 3*, Addison–Wesley, Reading, MA, 1973.
6. D.E. Knuth, *Mathematical analysis of algorithms*, Proc. of the 1971 IFIP Congress (1971), 19–27.
7. J. Lent and H. Mahmoud, *Average–case analysis of multiple quickselect: An algorithm for finding order statistics*, Statistics and Probability Letters (1995) (to appear).
8. M. Mahmoud, R. Modarres, and R. Smythe, *Analysis of Quickselect: an algorithm for order statistics*, RAIRO, Theoretical Informatics and Applications (1995) (to appear).
9. R. Serfling, *Approximation Theories of Mathematical Statistics*, Wiley, 1980.

TU Vienna
Wiedner Hauptstrasse 8–10
A-1040 Vienna
Austria
E-mail address: proding@rsmb.tuwien.ac.at