CIS 575. Introduction to Algorithm Analysis
Material for February 28, 2024

Linear Time Selection

The topic of this note is covered in *Cormen*'s Section 9.3.

1 Linear Selection by Clever Divide & Conquer

Recall that our goal is to find a deterministic **linear-time** algorithm for the **selection** problem, and that our approach is to choose a **pivot** which is then used by the Dutch National Flag algorithm, with a recursive call eventually made on (at most) one of the partitions.

Recall that our tentative idea was to use the **median** as the pivot, since if this can be done in linear time we would have the recurrence

$$T(n) \in T(\frac{n}{2}) + \Theta(n)$$

in which case we would indeed get the desired $T(n) \in \Theta(n)$. But we saw that this would be a "chicken-and-egg" situation since finding the median is a special case of the selection problem.

Instead, we shall explore the idea: rather than finding the exact median of all the numbers, we *find the median of a smaller sample*! To be specific, we

1. divide the numbers into *chunks of 5* (any remaining numbers will be ignored when finding the pivot)
2. for each chunk, compute its median
3. compute (to be used as the pivot) the median of all the medians.

We shall illustrate this approach on our example:

chunk	*median*
37, 22, 42, 11, 17	22
48, 12, 16, 20, 45	20
61, 24, 47, 53, 33	47
44, 35, 19, 10, 50	35
13, 16, 30, 54, 23	23
median of medians	**23**

Our method thus selects 23 as the pivot (we have already seen the resulting partitioning).

Running time The algorithm outlined above performs various tasks. To find a recurrence for the total running time $T(n)$, we shall now analyze the running time of each task:

1. For each 5-chunk, it computes its median. Computing the median of 5 elements can clearly be done in *constant* time (some methods may be faster than others but they all do a bounded number of comparisons). Hence the running time of this phase is in $\Theta(\frac{n}{5}) = \mathbf{\Theta(n)}$.

2. It computes the median of the $\frac{n}{5}$ sub-medians. This is done by a *recursive call* and thus takes time in $\mathbf{T(\frac{n}{5})}$.

3. It applies the Dutch National Flag Algorithm which takes time in $\mathbf{\Theta(n)}$.

4. It makes (at most) one recursive call, on a partition whose size could be more than $\frac{n}{2}$ but which we shall soon show cannot be more than qn for a certain q. The running time of this phase is thus at most $\mathbf{T(qn)}$.

We have justified the recurrence

$$T(n) = T(\frac{n}{5}) + T(qn) + \Theta(n)$$

but are left with the question:

how can we estimate q?

It turns out that $T(n) \in \Theta(n)$ whenever $T(n)$ is given by a recurrence of the form $T(n) = T(xn) + T(yn) + n$ where $x + y < 1$.

Hence we shall aim at finding a q with $q < 0.8$ such that no partition can be larger than qn. We may write n on the form $n = 5k + r$ with $0 \leq r < 5$. Then there will be k medians, and the pivot is chosen such that it is $\geq$ at least $\frac{k}{2}$ of those medians. But each of these medians is $\geq$ at least 3 numbers. Hence the pivot will be $\geq$ at least $3\frac{k}{2}$ numbers. That is, the fraction of numbers that are $\leq$ the pivot is at least

$$\frac{3\frac{k}{2}}{n} = \frac{3k}{10k + 2r}$$

which for big enough k is ≥ 0.29.

In other words, at most 71 % of the numbers will be $>$ the pivot; similarly, at most 71 % of the numbers will be $<$ the pivot. This shows (for big enough n) that with $q = 0.71$ (and thus $q < 0.8$ as required), each partition will be of size at most qn.

Concluding remarks We have seen that the selection problem allows a solution that (without any randomization) runs in linear time.

The solution is based on splitting the input into chunks of 5 elements. You may ask why we chose that number; what is so special about 5? Well, it should be clear that an even number would not be suitable, since for 4 elements, any median (the 2nd or the 3rd smallest) could induce a rather lopsided distribution (having chunks of 2 would be even worse).

But what about splitting into chunks of 3? It turns out that then the recurrence would be

$$T(n) \in T(\frac{n}{3}) + T(\frac{2n}{3}) + \Theta(n)$$

which does *not* allow us to prove $T(n) \in \Theta(n)$ (but only $T(n) \in \Theta(n \lg(n))$).

On the other hand, we could have chosen chunks of 7.