TB Schardl
6.046J/18.410J
September 25, 2009
Recitation 3

Deterministic Select

Problem: Given an unsorted set of n elements, find the ith **order statistic** of that set (the ith smallest element in the set.)

The obvious way to do this takes $O(n \log n)$ time. There is an efficient randomized way to do this in expected $O(n)$ time, which can be found in CLRS. We will see a way to do this deterministically in $O(n)$ time. (This algorithm is due to Blum, Floyd, Pratt, Rivest, and Tarjan.)

Note: By convention, when we discuss the median of a set with an even number of elements, we mean the "lower median" in that set. In other words, in a set of n elements where n is even, we take the median of that set to be the $\lfloor (n+1)/2 \rfloor$th element.

SELECT(A, i): where $n = |A|$

1. Divide the n elements of A into $\lfloor \frac{n}{5} \rfloor$ groups of 5 elements each. Additional elements may be placed in their own group of size $n \bmod 5$.

2. Find the median of each the groups. (Insertion-sort the elements in each group and then pick the median in each sorted list.)

3. Recursively SELECT the median x of the medians found in step 2.

4. Partition the input array around the median-of-medians x. Define $k = rank(x)$: k is one more than the number of elements on the low side of the partition, so x is the kth smallest element and there are $n - k$ elements on the high side of the partition.

5. If $i = k$, then return x. If $i < k$, recursively SELECT the ith smallest element on the low side. Otherwise $i > k$, so recursively SELECT the $(i-k)$th smallest element on the high side.

Claim: SELECT finds the ith order statistic of A in $O(n)$ worst-case time.
Proof.

We must evaluate the recurrence $T(n)$ for SELECT. Steps 1, 2, and 4 are non-recursive steps that take $O(n)$ time. Step 3 is a recursive call over $\lceil \frac{n}{5} \rceil$ elements - the median element from each group - which takes $T\left(\lceil \frac{n}{5} \rceil\right)$ time.

To evaluate the runtime of the recursive call in step 5, WLOG assume we must recurse on the elements larger than the median-of-medians, x. If we conceptualize the distribution of elements after step 3 of SELECT in the manner shown in Figure 1, we notice that the elements within the purple box are all known to be smaller than x. In general, if we have $\lceil \frac{n}{5} \rceil$ groups, then we have $\lfloor \frac{1}{2} \lceil \frac{n}{5} \rceil \rfloor$ groups whose median is at most x, and therefore $\lfloor \frac{1}{2} \lceil \frac{n}{5} \rceil \rfloor - 1$ groups whose median is less than x. Each of these groups contributes 3 elements that are less than x, so the number of elements less than x is at least

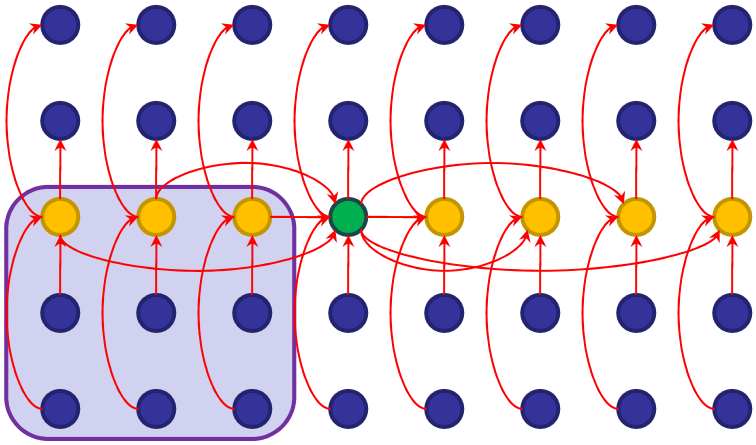

Figure 1: Conceptual layout of elements after step 3 of SELECT. The n elements are represented by circles, and each group of 5 is sorted in a column. The median of each group is shown in yellow, while the median of those medians, x, is shown in green. Arrows point from smaller elements to larger elements. The purple box highlights a subset of the elements known to be smaller than x.

$$\begin{aligned} 3\left(\left\lfloor \frac{1}{2}\left\lceil \frac{n}{5}\right\rceil \right\rfloor -1\right) &\geq 3\left(\left\lfloor \frac{n}{10}\right\rfloor -1\right) \\ &\geq 3\left(\frac{n}{10}-2\right) \\ &= \frac{3n}{10}-6 \end{aligned}$$

Therefore there are at most $\frac{7n}{10}+6$ elements greater than x, so the recursive call in step 5 takes at most $T(7n/10+6)$ time. The total runtime of SELECT is therefore

$$T(n) \leq T\left(\lceil n/5\rceil\right) + T(7n/10+6) + O(n).$$

We will use the substitution method to verify that this procedure runs in $O(n)$ time. To do this we will replace the $O(n)$ term in our recurrence with a representative function, an for sufficiently large a. We will also assume that for all $n < 140$ this method requires $O(1)$ time. To perform the substitution, assume $T(k) \leq c * k$ for some sufficiently large c and all $k > 0$. Substituting this inductive hypothesis into our recurrence gives:

$$\begin{aligned} T(n) &\leq c\lceil n/5\rceil + c(7n/10+6) + an \\ &\leq cn/5 + c + 7n/10 + 6c + an \\ &= 9cn/10 + 7c + an \\ &= cn + (-cn/10 + 7c + an) \end{aligned}$$

This is at most cn if $-cn/10 + 7c + an \leq 0$, which holds as long as $c \geq 10a(n/(n-70))$. Because we assume that for $n < 140$ this method runs in constant time, we find that $n/(n-70) \leq 2$, so choosing a $c \geq 20a$ will satisfy this inequality. Therefore SELECT runs in $O(n)$ time.