EECS 3101 M: Design and Analysis of Algorithms

Suprakash Datta
Office: LAS 3043

Course page: `http://www.eecs.yorku.ca/course/3101M`
Also on Moodle

Computing Order Statistics

Order statistics: The i^{th} order statistic of n elements $a_1, a_2, \ldots, a_n$ is the i^{th} smallest element

- Special cases: Minimum and maximum, Median
- How do we find the i^{th} order statistic?

Already seen:

- $k = 1$: $\Theta(n)$ algorithm, optimal
- Also, Build-Heap + Extract-max: $\Theta(n)$ algorithm
- Same bounds hold for any constant k
- Sorting solves it for any k: $\Theta(n \log n)$ algorithm
- What about $k = n/2$? Can we do better than $\Theta(n \log n)$?

Computing Order Statistics - 2

To select the i^{th} order statistic:

- Can we use PARTITION?
- if we are very lucky, we will get it in the first try!
- otherwise we should have a smaller set to recurse on.
- No guarantee of being lucky!
- How can we guarantee a significantly smaller set?

The algorithm is the most complicated divide-and-conquer algorithm in this course!

Computing Order Statistics: Algorithm SELECT

- Divide n elements into $\lceil n/5 \rceil$ groups of 5 elements.
- Find the median of each group.
- Use SELECT recursively to find the median x of the above $\lceil n/5 \rceil$ medians.
- Partition using x as pivot, and find position k of x
- If $i = k$ return
- else recurse on the appropriate subarray.

What kind of split does this produce?

Algorithm SELECT - 2

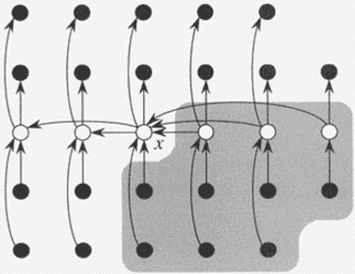

Figure 9.1 Analysis of the algorithm SELECT. The n elements are represented by small circles, and each group occupies a column. The medians of the groups are whitened, and the median-of-medians x is labeled. (When finding the median of an even number of elements, we use the lower median.) Arrows are drawn from larger elements to smaller, from which it can be seen that 3 out of every full group of 5 elements to the right of x are greater than x, and 3 out of every group of 5 elements to the left of x are less than x. The elements greater than x are shown on a shaded background.

SELECT - Analysis

- Steps 1,2,4 take $O(n)$
- Step 3 takes $T(\lceil n/5 \rceil)$
- Step 5: At least half of medians in step 2 are $\geq x$, thus at least $\lceil 1/2 \lceil n/5 \rceil \rceil - 2$ groups contribute 3 elements which are $\geq x$
  i.e, $3(\lceil 1/2 \lceil n/5 \rceil \rceil - 2) \geq (3n/10) - 6$
- Similarly, the number of elements $\leq x$ is also at least $(3n/10) - 6$
  Thus, $|S_1|$ is at most $(7n/10) + 6$, similarly for $|S_3|$
- Thus SELECT in step 5 is called recursively on at most $(7n/10) + 6$ elements.

SELECT - Recurrence

Recurrence is:

$$
\begin{aligned}
T(n) &= O(1) \text{ if } n < 140 \\
&= T(\lceil n/5 \rceil) + T(7n/10 + 6) + O(n) \text{ if } n \geq 140
\end{aligned}
$$

Show $T(n) \leq cn$, for some $c > 0$

SELECT - 2

$$
\begin{aligned}
T(n) &\leq c\lceil n/5 \rceil + c(7n/10 + 6) + an \\
&\leq cn/5 + c + 7/10cn + 6c + an \\
&= 9/10cn + an + 7c \\
&= cn + (-cn/10 + an + 7c) \\
&\leq cn \text{ if} \\
-cn/10 + an + 7c &< 0 \\
c &\geq 10a(n/(n-70)) \text{ when } n > 70
\end{aligned}
$$

So select $n = 140$, and then $c \geq 20a$

Note: n need not be 140, any integer > 70 is OK

Exercise

Describe an $O(n)$ algorithm that, given a set S of n distinct numbers and a positive integer $k \leq n$, determines the k numbers in S that are closest to the median of S.