Order Statistics

Today:
- Median
- Order Statistics

Order Statistics

Can find some kinds of elements in $O(n)$ time.

Example: Minimum

```
MINIMUM(A)
1 min ← A[1]
2 for i ← 2 to length[A]
3 do if min > A[i]
4 then min ← A[i]
5 return min
```

Same for Maximum.

What about general selection problem?

Divide and conquer

To select an element with rank $i$:

- partition the array according to some $x$
  (as in Quicksort), assume $k$ elements $\le x$
- if $i > k$ recurse on the right side, otherwise recurse on the left side

```
A

k

p

\le A[r]

r

\ge A[r]

q
```
Option 1: choose $x$ at random

```
RANDOMIZED-SELECT(A,p,r,i)
1  if $p = r$
2    then return $A[p]$
3  $q \leftarrow$ RANDOMIZED-PARTITION($A,p,r$)
4  $k \leftarrow q - p + 1$\> What is $k$?
5  if $i \leq k$
6    then return
7      RAND-SELECT($A,p,q,i$)
8  else return
9      RAND-SELECT($A,q+1,r,i-k$)
```

Example

Find 4th element of
A: 4 7 2 9 8
A: 9 8
A: 8
A: 8

Analysis

Very lucky (split in the middle):

$$T(n) \leq T\left(\frac{1}{2}n\right) + \Theta(n)$$

$$= \Theta(n)$$

Unlucky:

$$T(n) = T(n-1) + \Theta(n)$$

$$= \Theta(n^2)$$

Worst case is worse than sorting!

Analysis Cont.

Recall from the Quicksort lecture:

- a split (say, of $A[p \ldots r]$) is “lucky” for $a_i$, if the part where $a_i$ ends up in is of size $\leq \frac{3}{4}(r-p+1)$, i.e., if we reduce the array size by a factor of $\frac{3}{4}$
- after $\log\frac{4}{3}n$ lucky splits, we are done
- at any time, $a_i$ is lucky with probability at least $\frac{1}{4}$
Analysis Cont.

Let $T_i$ be the number of partitions between the $i - 1$th and $i$th lucky split.
The total time is bounded by

$$T = \sum_i T_i(3/4)^{i-1}n = n\sum_i T_i(3/4)^{i-1}$$

Therefore, the expected running time is at most

$$E[T] = E[n\sum_i T_i(3/4)^{i-1}] = n\sum_i (3/4)^{i-1}E[T_i]$$

(by linearity of expectation).

Analysis cont.

What is $E[T_i]$?

- sequence of independent trials with probability $p \geq 1/4$ of success
- $T_i = T_1$ is the waiting time for the next success

Turns out $E[T_1] = 1/p$.

Proof: We can write

$$E[T_1] = p \cdot 1 + (1 - p) \cdot (1 + E[T_1])$$

since with probability $p$ we are done, and with probability $1 - p$ we used one trial and have to continue.
The theorem follows.

Analysis - the final step

$$E[T] = n\sum_i (3/4)^{i-1}E[T_i] = n\sum_i (3/4)^{i-1} = O(n)$$

(since $(3/4)^{i-1}$ forms a geometric sequence).

Done!

(linear)

(but the algorithms is randomized)

Linear-Time Selection

Running time linear in worst case

Of theoretical interest only

Idea: generate a good partitioning element $x$. 
Select Algorithm

Select(\(i\))
1. Divide the \(n\) elements into groups of 5.

2. Find median of each group of 5 by sorting (constant time!).

3. Use Select recursively to find median \(x\) of \([n/5]\) medians from Step 2

Select Algorithm
Partition elements around \(x\). Let \(k = \text{rank}(x)\).

\[
\begin{array}{c|c|c}
\end{array}
\]

if \(i = k\)
then return \(x\)

if \(i < k\)
then use Select recursively to find \(i\)th smallest in low part
else use Select recursively to find \((i - k)\)th smallest in high part
Fractional Ordering

\[ \text{Analysis} \]

At least \(1/2\) of 5-element medians \(\leq x\).
At least \([n/5]/2 = [n/10]\) medians are \(\leq x\).
At least \(3[n/10]\) elements \(\leq x\).
For \(n \geq 50\), \(3[n/10] \geq n/4\).

\[ \text{Analysis} \]

So, after partitioning around \(x\), step 5 is called on \(\leq 3n/4\) elements, so
\[ T(n) \leq T(n/5) + T(3n/4) + O(n) \]
Looks like linear time, since \(1/5 + 3/4 < 1\).

Formally, substitute \(T(n) \leq cn\)
\[ T(n) \leq cn/5 + 3cn/4 + O(n) \]
\[ = 19cn/20 + O(n) \]
\[ = cn - (cn/20 - O(n)) \]
\[ \leq cn \text{ if } c \text{ big enough} \]
\(\Rightarrow\) \text{SELECT} is deterministic, and linear time!

\[ \text{Intuition} \]

- work at each level of recursion is reduced by a constant fraction
- analysis involves geometric sum
- work at root dominates \(\Rightarrow\) linear time
Applications

Worst-case $\Theta(n \log n)$ quicksort
1. Find median $x$ and partition
2. Recursively sort two halves.

$T(n) = 2T(n/2) + \Theta(n) = \Theta(n \log n)$

With worst case linear-time median, many problems reduce to simple divide and conquer methods!