Finding k^{th} smallest element

Section 9.2 CLRS

Input: An array of n real numbers $A[1 \ldots n]$ and $k \in \mathbb{N}$, $k \leq n$

Output: k^{th} smallest of A

Assume distinct. We could break ties by saying $A[i] < A[j]$ if $A[i] = A[j]$ and $i < j$

Solution 1 (a reduction)

Use sorting to solve the problem.

$A = \text{Sort}(A)$
return $A[k]$

+ foolproof $O(n \log n)$
− $\Omega(n \log n)$

Solution 2

Build a heap H from A ← $O(n)$
Do k times : $x = \text{extractMin}(H)$ ← $O(k \cdot \log n)$
return x

+ foolproof $O(n + k \log n)$
− $\Omega(n + k \log n)$ can be as bad as $\Omega(n \log n)$

Solution 3 (recursion)

Idea: (Quicksortish). split A into small (S) and large (L) elements. Compare $|S|$ to k to choose whether to look in S or L.

RSelect (A, k)
  If $|A| = 1$ return $A[1]$
  Pick pivot p from $A[1 \ldots n]$ at random (each element equally likely)
  $S = \{ A[i] \mid A[i] < p \}$
  $L = \{ A[i] \mid A[i] > p \}$
  If $|S| \geq k$ return RSelect (S, k)
  else if $|S| = k-1$ return p
  else return RSelect $(L, k - |S| - 1)$

If we could guarantee that the pivot is "in the middle" then the recursive subproblem will be small

If pivot has $\leq (1-\varepsilon)n$ elements smaller than it
and $\leq (1-\varepsilon)n$ elements bigger than it
then the size of the problem decreases by a factor $\leq (1-\varepsilon)$.

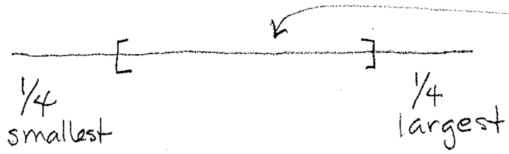
In that case the running time $T(n)$ would be

Assume the worst value of k so that we recurse in the larger of the two possible subproblems.

$$T(n) \leq c \cdot n + T((1-\varepsilon)n)$$
$$\leq c \cdot n + c(1-\varepsilon)n + c(1-\varepsilon)^2 n + \dots$$
$$\leq c \cdot n \underbrace{\left(1 + (1-\varepsilon) + (1-\varepsilon)^2 + \dots\right)}_{\text{converging geometric series}}$$
$$\leq c \cdot n \, 1/\varepsilon$$

A randomly chosen pivot will be "in the middle" often enough to cause the problem size to decrease rapidly.

Suppose we want a pivot that has $\leq 3/4\, n$ elements smaller
and $\leq 3/4\, n$ elements larger



any pivot in here is "good"

Probability that we choose a good pivot is $1/2$

$\Rightarrow$ Expected number of choices before we pick a good pivot is at most 2

random variable

Remember: Let X = # fair coin tosses before head

expected value of X

$$E[X] = \sum_{x=1}^{\infty} x \Pr[X = x]$$

definition

$$= \sum_{x=1}^{\infty} x \left(1/2\right)^x = 1/2 + 2/4 + 3/8 + 4/16 + \dots$$

$$2E[X] = 1 + 2/2 + 3/4 + 4/8 + \cdots$$

$$2E[X] - E[X] = 1 + 1/2 + 1/4 + 1/8 + \cdots$$
$$= 2$$

$$\Rightarrow \quad E[X] = 2$$

Notice that if we fail to choose a good pivot, the probability of choosing a good pivot in the next step increases.

Let X_i = time spent by RSelect when A is of size between $n(3/4)^i$ and $n(3/4)^{i+1}$

$Y = X_0 + X_1 + X_2 + \ldots$ is the running time of RSelect.

$$E[X_i] \leq 2\, \underbrace{c\,\underbrace{n\,(3/4)^i}_{|A| \text{ is at most this}}}_{\text{time spent partitioning}}$$

linearity of expectation
$E[A+B]$
$= E[A] + E[B]$

$$\Rightarrow E[Y] = E[X_0] + E[X_1] + E[X_2] + \cdots$$
$$\leq \sum_{i=0}^{\infty} 2cn(3/4)^i$$
$$= 8cn$$

\+ foolproof, expected running time $= O(n)$

\- randomized ... sometimes RSelect might take a long time.

Solution 4   Deterministic Linear time k-Select

Idea Deterministically pick a good pivot p.

by Blum, Floyd, Pratt, Rivest, Tarjan 1972

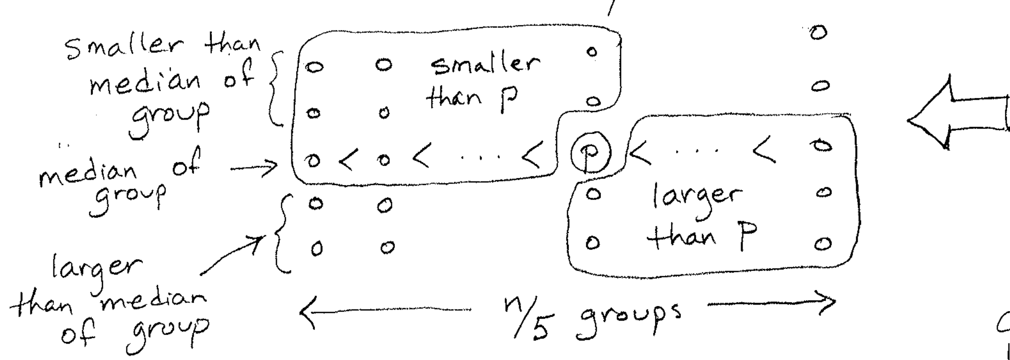
BSelect (A, k)

If $|A| = 1$ return $A[1]$
$\boxed{p = \text{GoodPivot}(A)}$
$S = \{A[i] \mid A[i] < p\}$
$L = \{A[i] \mid A[i] > p\}$
If $|S| \geq k$ return BSelect(S, k)
else if $|S| = k-1$ return p
else return BSelect$(L, k - |S| - 1)$

Good Pivot (A)

1. Divide A into $n/5$ groups of 5 elements each.
2. Find median of each group of 5.
3. Use BSelect to find the median, p, of the $n/5$ medians from step 2
4. return p.

At least how many elements of A are $< p$?



Arrange elements of A into this picture. Each element is a o.

Note: the algorithm doesn't do this. We do it to see how many elements are $<p$.

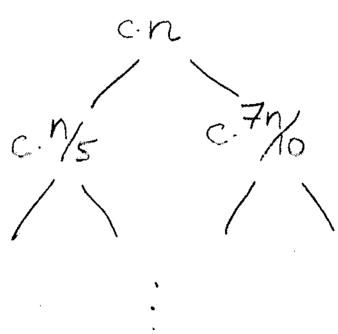
elts. smaller than $p \geq 3(n/10) - 1 \Rightarrow |S| \geq \frac{3n}{10} - 1 \Rightarrow |L| \leq 7n/10$

elts. larger than $p \geq 3(n/10) - 1 \Rightarrow |L| \geq \frac{3n}{10} - 1 \Rightarrow |S| \leq \frac{7n}{10}$

Running time of BSelect: $T(n)$

- pick good pivot (medians of $n/5$ groups of size 5 $= O(n)$
  + BSelect median of $n/5$ medians $= T(n/5)$
- partition $= O(n)$
- recurse $= O(7n/10)$ $\Rightarrow$ $\boxed{T(n) \leq c \cdot n + T(n/5) + T(7n/10)}$

Recursion Tree for $T(n) = c \cdot n + T(n/5) + T(7n/10)$



$c \cdot n$

$c \cdot 9/10\, n$

$c \cdot (9/10)^2 n$

$$\text{Total} \leq c \cdot n \sum_{i=0}^{\log_{10/7} n} (9/10)^i \leq c \cdot n \sum_{i=0}^{\infty} (9/10)^i = 10cn$$