The Selection Problem

Given a list of n distinct comparable elements (numbers, say) and a number k with $1 \le k \le n$.

[Return k^{th} smallest element of the list.
k^{th} order statistic

First try. Select(A, k)
$\Theta(n \lg n)$ - Heapsort $A[1 \ldots n]$
$O(1)$ Return $A[k]$.

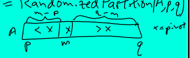
[might be good if multiple calls to select on same list.]

One call still requires $n \lg n$ overhead.

Can we do better? Yes
- Randomized select algo with $\Theta(n)$ worst-case expected time.
- Deterministic select algo in $\Theta(n)$ worst-case time.

RandSelect(A, p, q, k)
// return k^{th} smallest element
// in $A[p..q]$ $\quad p \le q$
// Preconditions: $1 \le k \le q-p+1$
// elements of A are distinct

if $p = q$:
    return $A[p]$
else:
    $m :=$ RandomizedPartition(A, p, q)



    if $k = m-p+1$:
        return $A[m]$
    if $k < m-p+1$:
        return RandSelect$(A, p, m-1, k)$
    else: // $k > m-p+1$
        return RandSelect$(A, m+1, q, k-(m-p+1))$

Analyze worst-case expected time $E(n)$ of RandSelect on n elements.

$$E(n) = an + \sum_{q=0}^{n-1} \Pr\left[\begin{array}{c}q \text{ is the length of}\\ \text{the left sublist}\end{array}\right] \cdot \left(\max\{E(q), E(n-q-1)\}\right)$$

const. a
an is partition time

$$= an + \frac{1}{n} \sum_{q=0}^{n-1} \max\{E(q), E(n-q-1)\}$$

Show that $E(n) = \Theta(n)$ by the substitution method. [in the Book].

Selection in deterministic worst-case time $\Theta(n)$:

Lemma: Let $T(n)$ satisfy

$$T(n) = T(\alpha n) + T(\beta n) + n$$

where $\alpha, \beta > 0$ are constants such that $\alpha + \beta < 1$.

Then $T(n) = \Theta(n)$.

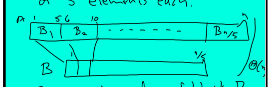
Upper bound: Assume $T(m) \le cm$ for all $m < n$
WTS $T(n) \le cn$.

$$T(n) = T(\alpha n) + T(\beta n) + n$$
$$\le c\alpha n + c\beta n + n$$
$$= n(c(\alpha + \beta) + 1)$$
$$\le cn \text{ provided}$$
$$c(\alpha+\beta) + 1 \le c$$
$$1 \le c(\underbrace{1-\alpha-\beta}_{>0})$$
$$\frac{1}{1-\alpha-\beta} \le c. \quad //$$

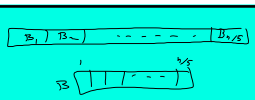
Now the algorithm:

Given $A[1 \ldots n]$

Chop A into block $B_1, B_2, \ldots, B_{n/5}$ of 5 elements each.



$B[j]$ is the median of block B_j



$B[j]$ is median of block B_j

Computing B takes time $\Theta(n)$,

Find median ($\frac{n}{10}$th order stat) of the B array by a recursive selection call on B.

Let this median be x.

Partition $A[1..n]$ using x as the pivot.

Call recursively on the sublist containing the order stat (as with RandSelect),

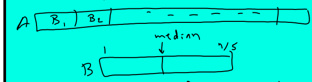
Let $T(n)$ be the worst-case time for this algo on n items.

$$T(n) = \underbrace{\Theta(n)}_{\text{medians of blocks, partition}} + \underbrace{T\left(\frac{n}{5}\right)}_{\text{time to recurse on the B-array}} + \underbrace{T(?)}_{\text{time to recurse on the worst-case sublist after partition of A}}$$

Assume $T(n)$ is monotone ascending in n

$$\to T(n) \leq n + T\left(\frac{n}{5}\right) + T(S)$$

where S is the biggest possible size of a sublist after partition.



$n/10$ elements of B less than the median m

$m_1, \ldots, m_{\frac{n}{10}} < m$

For all $1 \leq j \leq \frac{n}{10}$ let B be the block of A that contained m_j

B has 2 elements $< m_j$

so there are 3 elements in each block that are $< m$ "x".

$\geq 3 \cdot \left(\frac{n}{10}\right)$ many elements in A that are $< x = m$

Thus there are $\leq \frac{7n}{10}$ elements in A that are bigger than x.

By a symmetrical argument there are $\leq \frac{7n}{10}$ elements of A that are $< x$.

$\therefore$ Size of the worst-case of the sublist of A is $\frac{7n}{10}$

So,

$$T(n) \leq n + T\left(\overset{\alpha n}{\frac{n}{5}}\right) + T\left(\overset{\beta n}{\frac{7n}{10}}\right)$$

$$\alpha = \frac{1}{5} = \frac{2}{10}$$

$$\beta = \frac{7}{10}$$

$$\alpha + \beta = \frac{9}{10} < 1$$

Thus $T(n) = \Theta(n)$ by the lemma.

Median-of-5 approach given a deterministic QuickSort;

Find the median in deterministic linear time using median-of-5 select

Use this as the pivot & proceed as usual.