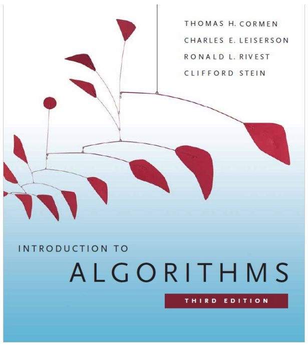
Order Statistics

Prepared by Shmuel Wimer

Courtesy of Prof. Dror Rawitz



Order Statistics and Selection Problem

The ***ith order statistics*** of n elements is the ith smallest element.

Given a set A of n distinct elements, the ith order statistics **selection problem** is to find $x \in A$ s.t. x is larger than exactly $i - 1$ elements.

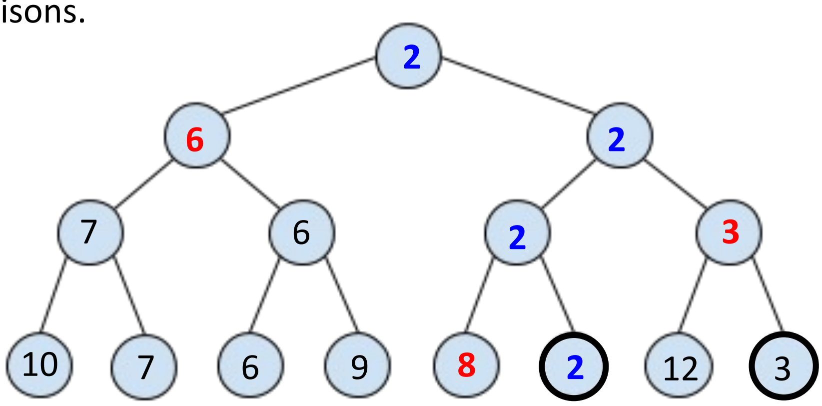
Sorting A in $O(n \log n)$ time enables to solve in $O(n)$ time, but faster solution in $O(n)$ time (no sorting) is possible.

Minimum $i = 1$ and **maximum** $i = n$ problems complexity is $\Theta(n)$.

$\Theta(n)$ is possible for any ith order statistics.

Finding the 2nd Smallest

Can be found by knockout tournament with $n + \lceil \log n \rceil - 2$ comparisons.



3

1. Divide input into $k = \lceil n/2 \rceil$ element pairs.
2. Get the set of k smaller of every pair.
3. If $k == 1$ stop, smallest found. Else set $n = k$ and go back to 1.

Knockout tournament has $n - 1$ comparisons.

Smallest wins after $\lceil \log n \rceil$ comparisons, implying binary tree.

At some comparison **the smallest must beat** the 2nd smallest.

Traversal of the $\lceil \log n \rceil - 1$ beats of smallest discover the 2nd.

4

Selection of kth Order Statistics in Expected $\Theta(n)$

A divide-and-conquer algorithm similar to Quicksort, except that it proceeds with only one of the partitions rather than both.

$\text{RANDOMIZED}-\text{SELECT}$ seeks the ith order statistics of A.

It partitions the array recursively.

It chooses a pivot randomly with $\text{RANDOMIZED}-\text{PARTITION}$ but proceed with only one side of the partition where i is surely located.

$\text{RANDOMIZED}-\text{SELECT}(A, p, r, i)$ // ith order statistics
    if $p == r$ **return** $A[p]$ // $A[p]$ ith smallest element
    // partition $A[p..r]$ into $A[p..q-1]$ and $A[q+1..r]$
    // around pivot $A[q]$
    $q = \text{RANDOMIZED}-\text{PARTITION}(A, p, r)$
    $k = q - p + 1$ // number of elements in $A[p..q]$
    if $i == k$ **return** $A[q]$ // $A[q]$ is ith smallest element
    else if $i < k$ // ith is in the lower part
        return $\text{RANDOMIZED}-\text{SELECT}(A, p, q-1, i)$
    else // ith is in the higher part
        return $\text{RANDOMIZED}-\text{SELECT}(A, q+1, r, i-k)$

Run time analysis

Worst-case run time is $\Theta(n^2)$. When it happens? (HW)

Run time $T(n)$ on $A[p..r]$ of n elements is random variable.

$\text{RANDOMIZED–PARTITION}(A, p, r)$ likely returns any element as pivot $A[q]$. $\Rightarrow$ $\Pr[|[p..q]| = k] = 1/n$, $1 \le k \le n$.

Define random variable $X_k = \begin{cases} 1 & |[p..q]| = k \\ 0 & \text{otherwise} \end{cases} \Rightarrow \mathrm{E}[X_k] = \frac{1}{n}$.

It is unknown a priori whether selection proceeds with $A[p..q-1]$, $A[q]$ or $A[q+1..r]$, so larger interval is safely assumed.

When $X_k = 1$, the implied subarrays have size $k-1$ and $n-k$. $\Rightarrow$ $T[n] \le \sum_{k=1}^{n} X_k \cdot \Big(T\big(\max(k-1, n-k)\big) + O(n)\Big)$.

Taking expectation and applying its linearity, there is

$$\mathrm{E}\big[T[n]\big] \le \sum_{k=1}^{n} \mathrm{E}\big[X_k \cdot T\big(\max(k-1, n-k)\big)\big] + O(n)$$

Independence: $\Pr[X_k = 0\ /\ 1 | \max(k-1, n-k)] = \Pr[X_k = 0\ /\ 1]$

$$= \sum_{k=1}^{n} \mathrm{E}[X_k] \cdot \mathrm{E}\big[T\big(\max(k-1, n-k)\big)\big] + O(n)$$

$$= \frac{1}{n} \sum_{k=1}^{n} \mathrm{E}\big[T\big(\max(k-1, n-k)\big)\big] + O(n).$$

There is $\max(k-1, n-k) = \begin{cases} k-1 & \text{if } k > \lceil n/2 \rceil \\ n-k & \text{if } k \leq \lceil n/2 \rceil \end{cases}$.

In $\sum_{k=1}^{n} \mathrm{E}[\cdot]$, for even n each term from $T(\lceil n/2 \rceil)$ to $T(n-1)$ appears twice. Same for odd n plus $T(\lfloor n/2 \rfloor)$ which appears once. $\Rightarrow$

$\mathrm{E}\big[T[n]\big] \leq \frac{2}{n} \sum_{k=\lfloor n/2 \rfloor}^{n-1} \mathrm{E}[T(k)] + O(n)$.

Above recursion is solved by substitution, yielding $\mathrm{E}[T(n)] = O(n)$. (HW, see CLRS 9.2).

Selection of kth Order Statistics in $O(n)$ Worst Case

$\mathrm{SELECT}(A, k)$

1. Divide input into $\left\lceil \frac{n}{5} \right\rceil$ groups of 5 elements, one possibly smaller.
2. Find the median of each group. Let B be the set of all medians.
3. Let $x = \mathrm{SELECT}\left(B, \left\lfloor \frac{|B|}{2} \right\rfloor\right)$ be B's median.
4. Divide A around x : $L = \{a \in A \mid a < x\}$, $R = \{a \in A \mid a > x\}$.
5. If $k = |L| + 1$ return x.

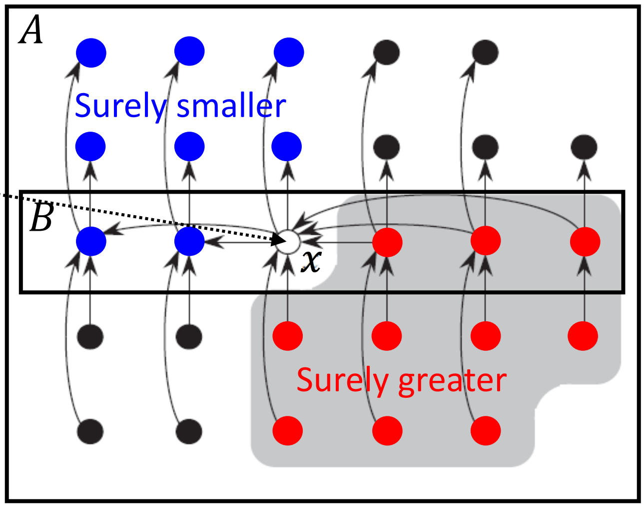
   $k > |L| + 1$
6. If $k \leq |L|$ then $\mathrm{SELECT}(L, k)$ else $\mathrm{SELECT}\big(R, k - (|L| + 1)\big)$.

●→●

greater to smaller

x: $\lfloor |B|/2 \rfloor$

$|L| \geq \frac{n}{4}$ and $|R| \geq \frac{n}{4}$ guaranteed.



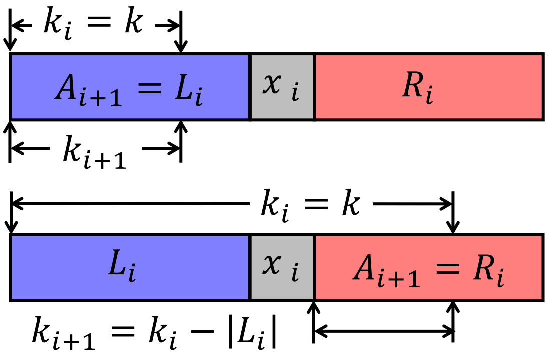
Correctness proof:

Consider $\mathrm{SELECT}(A_i, k_i)$ call at line 6, where $A_0 = A$ and $k_0 = k$. Assume by induction that $k_i = k$ and consider (A_{i+1}, k_{i+1}). Then

If $k_i \leq |L_i|$

then $k_{i+1} = k_i = k$

else $k_{i+1} = k_i - (|L_i| + 1)$



Run time analysis: Assume w.l.o.g that all the elements are distinct.

Then at least half of the medians in step 2 are not smaller than x.

$\Rightarrow$ At least half of the $\left\lceil \frac{n}{5} \right\rceil$ groups contribute 3 elements $> x$, maybe except one smaller group and the group containing x.

$\Rightarrow |R| \geq 3\left(\frac{1}{2}\left\lceil \frac{n}{5} \right\rceil - 2\right) \geq \frac{3n}{10} - 6.$ Similarly, $|L| \geq \frac{3n}{10} - 6.$

Consequently, $\frac{3n}{10} - 6 \leq |L|, |R| \leq \frac{7n}{10} + 6.$

Let $T(n)$ be the worst-case run time. Steps 1, 2 and 4 take $O(n)$ time.

Steps 3 takes $T(\lceil n/5 \rceil)$ time.

Step 6 takes at most $T(7n/10 + 6)$ time.

The following recurrence is in order

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

We show that $T(n) = O(n)$.

Assume $T(n) \leq cn$ for sufficiently large n and a constant c. Then

$T(n) \le c(\lceil n/5 \rceil) + c(7n/10 + 6) + an$ $O(n) = an$

$\le c(n/5 + 1) + c(7n/10 + 6) + an$

$= 9cn/10 + 7c + an$

by assumption

$= cn + (-cn/10 + 7c + an) \le cn$

$\Rightarrow \; -cn/10 + 7c + an \le 0$

$\Rightarrow$ (1) $c \ge 10a(n/(n-70)) \to 10a$ as $n \to \infty$.

Choosing $c \ge 20a$ will satisfy (1). ■

What happens if 5 is replaced by 3? 7? (HW).

SELECT is a deterministic algorithm. We can use probabilistic pivot choice in line 3 (as in Quicksort) and then proceed in one side of the partitions.

Show that its expected run time is linear (HW).

Quicksort can use SELECT for balanced partition, yielding worst-case $O(n \log n)$ time deterministic Quicksort.

Impractical because of the large constants in SELECT.