MEDIANS (OR SELECTION)

INPUT : A list of n-real numbers S, an integer k.

OUTPUT : The k^{th} Smallest element of S, for Some $1 \le k \le n$.

e.g $k=1 \Rightarrow$ MINIMUM and can be computed in $O(n)$ time.

$k=n \Rightarrow$ MAXIMUM and can be computed in $O(n)$ time.

$k = \lceil \frac{n}{2} \rceil \Rightarrow$ MEDIAN ELEMENT

How fast we can compute Median?
or in general the k-th Smallest.

Easy Solution: SORT(A)
OUTPUT k^{th}-element in the sorted list.

This takes $O(n \log n)$ time.

OBJECTIVE: k^{th} Smallest can be computed in $O(n)$ time.

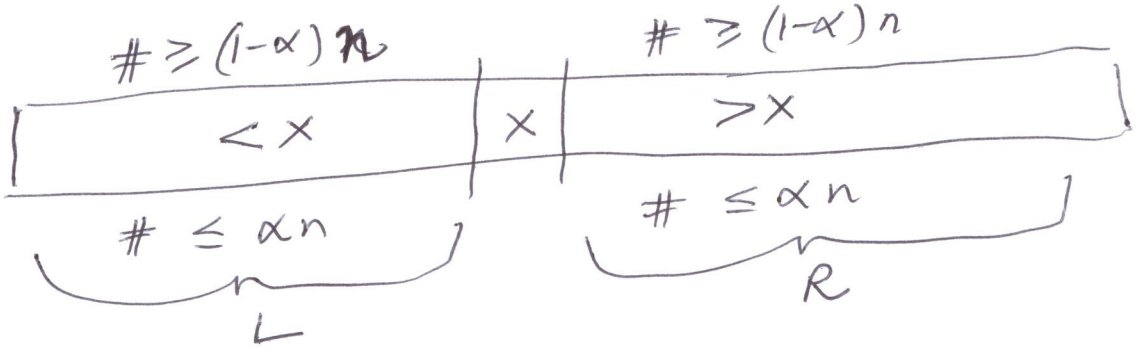
1973 BLUM
FLOYD
PRATT
RIVEST
TARJAN

Time Bounds for Selection
(or Median of Median algorithm)
based on Diride and Conquer.

Selection in Linear Time

Approach: In $O(n)$ time find an element x in S, which partition S as follows.

FIGURE 1



Here,

① Assume that all elements of S are distinct (this is to make explanation easier)

② Set L consists of all elements in S that are $< x$.

③ Set R consists of all elements in S that are $> x$.

④ α is a constant between $\frac{1}{2} < \alpha < \frac{3}{4}$.

We say that x partitions S in a BALANCED WAY.

Suppose we can find such an x that partitions S in a Balanced Way.

We are looking for k^{th} smallest element of S.

Question: ~~is~~ Where is the k^{th} smallest element? Is it $=x$, or in set L, or in set R.

Answer: That is easy to find.

(a) If $|L| \geq k$, then k^{th} element is the k^{th} element ~~is in set L~~ of set L.

(b) If $|L| = k-1$, then x is the k^{th} element.

(c) If $|L| < k-1$, then x is the ~~k~~ $(k - |L| - 1)^{th}$ element of set R.

$\Rightarrow$ Either we find the k^{th}-element or ~~our~~ we have to recursively find an appropriate element in either the set L or R.

If $T(n)$ = time to find the k^{th} smallest element then

$$T(n) = \underbrace{T(\alpha n)}_{\text{Since } |L| \leq \alpha n,\ |R| \leq \alpha n} + \underbrace{O(n)}_{\text{Time to find an element } x \text{ that partitions } S \text{ in a balanced way + Partitioning of } S \text{ in sets L and R}}$$

Then,

$$T(n) = \frac{1}{1-\alpha} n = O(n)$$

Since α is a constant.

What remains is to show that such an element x can be found in $O(n)$ time.

INPUT: Let $S = \{a_1, a_2, \ldots, a_n\}$ be n-numbers.

OUTPUT: k-th smallest element.

Assumption: Assume all elements of S are distinct (this is just to simplify the discussion)

Pseudo code

step1: Divide the input into $\lceil \frac{n}{5} \rceil$ groups, where each group contains 5-elements, but the last one may have fewer than 5.

$\underbrace{a_1\, a_2\, a_3\, a_4\, a_5}_{\text{group 1}}$, $\underbrace{a_6 a_7 \cdots a_{10}}_{\text{group 2}}$ $\underbrace{a_{11} a_{12} \quad a_{15}}_{\text{group 3}}$ ----- .....

Step2: For each group j, $\left(1 \le j \le \lceil \frac{n}{5} \rceil\right)$, Compute the median m_j of the j-th group.

[Since each group has 5-elements, we can sort them and compute its median in $O(1)$ time.]

Step3: x = median of $m_1, m_2, \ldots, m_{\lceil \frac{n}{5} \rceil}$.

↑ Thats what we were trying to find.

CLAIM At least $\frac{3n}{10}$ elements of S are $\leq x$.

Equivalently, at most $\frac{7n}{10}$ elements of S are $> x$.

Proof: Since x is median of $m_1, m_2, \ldots, m_{\lceil \frac{n}{5} \rceil}$,

then 50% of m_j's are $\leq x$.

$\Rightarrow \frac{n}{10}$ of m_j's are $\leq x$.

In each of the groups, where m_j came from, at least 3 elements in that group are $\leq x$.

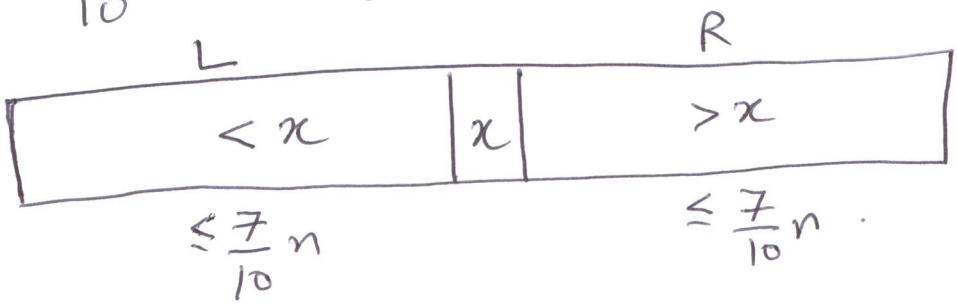
$\Rightarrow$ At least $\frac{3n}{10}$ elements in S are $\leq x$.

$\Rightarrow$ At most $\frac{7n}{10}$ elements in S are $> x$.

Corollary: At most $\frac{7n}{10}$ elements in S are $< x$.

Proof: Repeat the argument of the claim by using the fact that 50% m_j's are $\geq x$.

So $\alpha = \frac{7}{10}$ in Figure 1, for this value of x.



Step 4: Partition elements of S into L and R.

$L = \{u \mid u \in S$ such that $u < x\}$.

$R = \{u \mid u \in S$ such that $u > x\}$.

Step 5: If $|L| = k-1$: Return x

If $|L| \geq k-1$: Recursively, compute k^{th} smallest in L.

If $|L| < k-1$: Recursively, compute $(k-|L|-1)$-th smallest in R.

— x —

$$T(n) = O(n) \quad [\text{Step 1}]$$
$$+ O(n) \quad [\text{Step 2}]$$
$$+ T\left(\frac{n}{5}\right) \quad [\text{Step 3}]$$
$$+ O(n) \quad [\text{Step 4}]$$
$$+ T\left(\frac{7}{10}n\right) \quad [\text{Step 5}]$$

$$\Rightarrow T(n) = T\left(\frac{n}{5}\right) + T\left(\frac{7}{10}n\right) + O(n)$$

<u>claim</u> $T(n) = O(n)$.

<u>Proof</u>: By induction on n.

Base Case – Easy to verify.

Need to show

$$T(n) \leq T\left(\frac{n}{5}\right) + T\left(\frac{7}{10}n\right) + cn \overset{?}{\leq} dn.$$

$$\Leftrightarrow \underbrace{d\frac{n}{5}}_{\text{I.H}} + \underbrace{d\frac{7}{10}n}_{\text{I.H}} + cn \overset{?}{\leq} dn$$

$$\Leftrightarrow 10c \overset{?}{\leq} d$$

Since we can always find a constant $d = 10c$, claim holds.

Caution

Taking care of the fact that the last group may have anywhere from 1 to 5 elements, the true recurrence relation is

$$T(n) = \begin{cases} O(1) & \text{if } n \leq 140 \\ T(\lceil \frac{n}{5} \rceil) + T(\frac{7n}{10} + 6) + O(n) & \text{if } n > 140 \end{cases}$$

and by induction one can show that

$$T(n) \leq d\lceil \tfrac{n}{5} \rceil + d(\tfrac{7n}{10} + 6) + cn \overset{?}{\leq} dn \quad -(I)$$

$$\Leftrightarrow \quad d\tfrac{n}{5} + d + d\cdot\tfrac{7n}{10} + 6d + cn \overset{?}{\leq} dn$$

$$\Leftrightarrow \quad \tfrac{9dn}{10} + 7d + cn \overset{?}{\leq} dn$$

$$\Leftrightarrow \quad -\tfrac{d}{10}n + 7d + cn \overset{?}{\leq} 0$$

$$\Leftrightarrow \quad d \geq \frac{10cn}{n-70}, \quad \text{since } n \geq 140 \ \frac{n}{n-70} \leq 2$$

$\Rightarrow d \geq 20c$ will satisfy (I).

Are there other ways to find x.

What if we choose x randomly.

It is obvious that sometimes we may be unlucky (e.g $x = \text{MIN}$ or $x = \text{MAX}$) or we may be ~~un~~lucky (e.g x is an element that partitions S in a balanced way).

Call CHOICE of x GOOD if

the element x ~~leaves at~~ has at least $\frac{|S|}{4}$ elements $< x$ and at least $\frac{|S|}{4}$ elements $> x$.

~~(~~

What is the probability of picking such an element x?

Easy $= 1/2$

Why: Smallest 25% and Largest 25% elements are not good choice - except that

all elements are good.

$\Rightarrow$ Given a random x, it has 50% chance of being good.

$\Rightarrow$ Within two trials ~~we can find~~ on the average we can find a good x.
(= How many times we have to toss a coin till we get "Head")

Hence

$T(n)$ = Expected Running Time

$$= T\left(\frac{3}{4}n\right) + O(n)$$

Time to reduce the ~~area~~ set to $\frac{3}{4}$ of its size (including on the average two trials).

$$= O(n).$$

The above Expected Running Time Analysis was very sloppy.

We need to use linearity of expectation in a proper way.

In this course we will mainly stay with deterministic algorithms. ⊠