OH TODAY

- 1:30pm - 2:20pm
- Zoom

Exam 1

- Feb 27
- during class time
- seating will be posted on class page.

Selection

Input: Array A containing n distinct integers.

Obj: To find i^{th} smallest element in A.

24 36 4 $\boxed{18}$ 71

Naive:
- Sort the array.
- return the i^{th} element in the sorted array.

$O(n \lg n)$

Alg → Select (A, i)

1. Divide the array into $\lfloor n/5 \rfloor$ groups, each containing exactly 5 elements & one group containing $n \bmod 5$ elements. $O(n)$

2. Find the median of each of the groups. Call this set M. $\rightarrow$ $O(n)$

3. Find the median of the medians. Call this x. Recursively call the alg- Select $(M, \frac{1}{2}\lceil \frac{n}{5} \rceil)$
$T(\lceil \frac{n}{5} \rceil)$

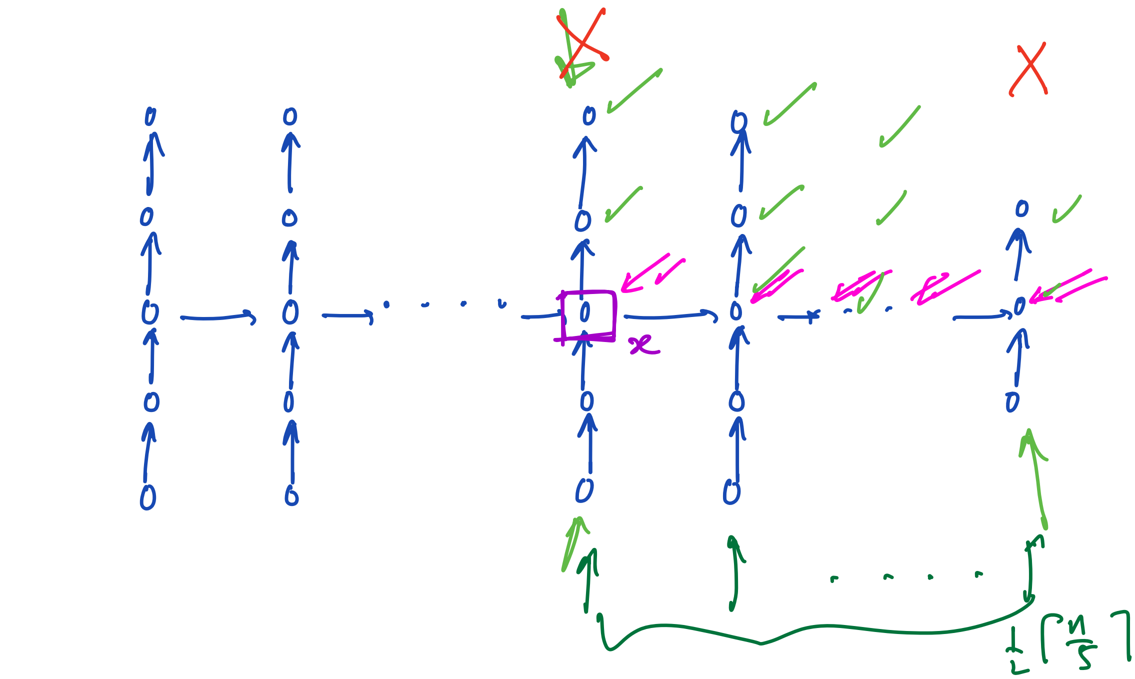
from QS

4. Partition the elements in A around x. Let k = rank(x). $\rightarrow$ $O(n)$
position of x in the sorted array.

5. if $k = i$ then return x. $\rightarrow$ $O(1)$

6. else if $k > i$ then return Select(A[1..k-1], i)

7. else return Select(A[k+1..n], i-k).



$\#$ elements $> x \;\geq\; 3\Big(\underbrace{\tfrac{1}{2}\left\lceil \tfrac{n}{5} \right\rceil - 2}_{\text{these many cols give us exactly 3 elems} > x.}\Big)$

$$\geq \frac{3n}{10} - 6.$$

In the worst case, we will recurse on the larger partition, which will contain $\leq \frac{7n}{10} + 6$ elements.

Runtime recurrence

$$T(n) \le \begin{cases} O(1), & n \le 140 \\ T\left(\left\lceil \frac{n}{5} \right\rceil\right) + T\left(\frac{7n}{10} + 6\right) + \underbrace{O(n)}_{a \cdot n}, & \text{o.w} \end{cases}$$

$a \cdot n$, where a is some const.

Target runtime: $\Theta(n)$

We will prove using induction that

$$\boxed{T(n) \le cn, \quad \forall n \ge n_0}$$

, for some positive consts c & n_0.

BC: $n \le 140$

$T(n) \le c \cdot n$, for sufficiently large c.

IH: Let k be an arb but particular integer $> n_0$. Assume that

$T(j) \le cj, \quad \forall n_0 \le j \le k-1$.

IS: We want to show that

$T(k) \le ck$.

$$T(k) \le T\left(\left\lceil \frac{k}{5} \right\rceil\right) + T\left(\frac{7k}{10} + 6\right) + ak$$

$$\le c \cdot \left\lceil \frac{k}{5} \right\rceil + c\left(\frac{7k}{10} + 6\right) + ak \quad \text{(By IH)}$$

$$\le c\left(\frac{k}{5} + 1\right) + c\left(\frac{7k}{10} + 6\right) + ak$$

$$= \frac{9k}{10} + 7c + ak$$

$$= ck + \left(-\frac{ck}{10} + 7c + ak\right)$$

RHS will be $\le ck$, if

$$7c + ak - \frac{ck}{10} < 0$$

$$7c + ak < \frac{ck}{10}$$

$$\therefore \quad c\left(\frac{k}{10} - 7\right) > ak$$

$$\therefore \quad c > \frac{10ak}{k - 70}$$

Set $k = \underline{\underline{140}}$

$$\therefore \quad \boxed{c > 20a.} \quad , \quad \boxed{n_0 = 140}$$

Stacks

- ADT (Abstract Data Type)
- abstraction

- elements
- operation on the elems.

For Stack — Push
— Pop

Array implementation.

Push (obj)

// s : stack size ; index starts at 0

// a : array size

// c : initial size & the increment.

A[s] ← obj

s++

if s == a then

a ← a + c

Alt

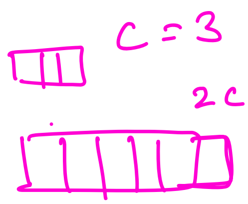
a ← 2 * a

copy contents into the new array.

Analysis : We will push n elements onto the stack.

$$T(n) = n + c + 2c + 3c + \cdots + n$$

$$= n + c\left(1 + 2 + \cdots + \frac{n}{c}\right)$$



$$= n + c \cdot \frac{\left(\frac{n}{c}\right)\left(\frac{n}{c}+1\right)}{2}$$

$$= \underline{\Theta(n^2)}$$

Alt implementation ($a \leftarrow 2 * a$)

Suppose initial size of the array is 1.

max # times we will need to resize

$= \lg n$

time to copy $\leq n$

$\therefore \quad T(n) = O(n \lg n).$

$= \Theta(n \lg n).$ X

$$T(n) = n + \underbrace{1 + 2 + 2^2 + \cdots + 2^{\lg n}}$$

$$= n + 2^{\lg n + 1} - 1$$

$$= n + 2 \cdot 2^{\lg n} - 1$$

$$= n + 2n - 1$$

$$= \Theta(n) \checkmark .$$