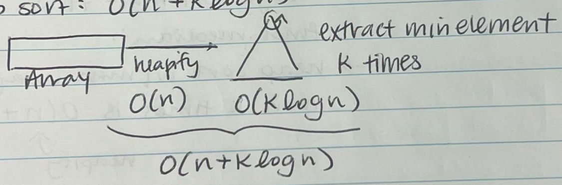
9/18/23

Today
- Selection Problem
- Algo & runtime
- Picking a pivot
  - M-O-M & Runtime
- Union-Find Data Struct for connected components (graph algorithms!!)
- Quiz Weds! 10-15 min. not mult choice, solve problems and explain. everything since 1st class up until selection. Understand all the algos, runtimes, including the HW.

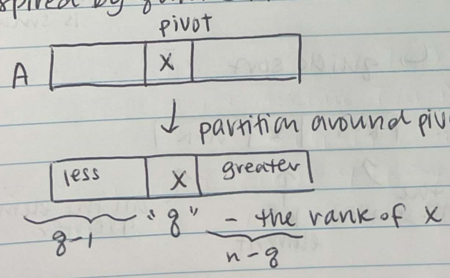
primitive operations

Selection problem: Given an array A of comparable elements, determine the k^{th} smallest element.
Assumption: elements are distinct.

- merge sort: $O(n \log n)$
- heap sort: $O(n + k \log n)$



- Inspired by quick sort:



if $q = k$, then x is the element to return
if $q > k$, search for k^{th} element in $A[1:q-1]$
if $q < k$, search for $(k-q)^{th}$ element in $A[q+1:n]$

1	2	3	4	5	6	7	8	9
5	8	2	6	4	1	3	7	9

$k = 6^{th}$ smallest

1	2	3	4	5	6	7	8	9
2	1	3	4	5	8	7	6	9

5	6	7	8	9
5	8	7	6	9

pivot is A[7]
$q = 7$

partition around A[7]

5	6	7	8	9
5	6	8	7	9
1	2	3	4	5

return 6

Algo: Selection (A, p, r, k)
Input: Array A of n distinct comparable elements
Output: k^{th} smallest element in A[p:r]

where: ~~$1 \le p \le q \le n$~~
$1 \le k \le r - p + 1$
$1 \le p \le r \le n$

* k changes while invoking

Base if $p = r$ then return A[p]
x ← choose pivot using M-O-M.
partition (A[p:r], x) $O(n)$
q ← find rank of x in A[p:r] $O(1)$
if $k = q$ then return A[q]
if $k < q$ then return selection (A, p, q − 1, k)
if $k > q$ then return selection (A, q+1, r, k − q)

selection $(A, 1, 9, 6)$ $A = \boxed{5|8|2|6|4|1|3|7|9}$

↳ selection $(A, 5, 9, 2)$

$\boxed{5|8|6|7|9}$

After partitioning around the pivot, search for ele. in this subarray. New $k=2$

<u>Picking a pivot</u>

Scenario: pick a pivot that leaves a subarray of 75% of elements that we use for the next invocation.

$\underbrace{\boxed{<}}_{\substack{25\% \\ 1/4\, n}}\boxed{x}\underbrace{\boxed{\quad >\quad}}_{\substack{75\% \\ 3/4\, n}}$ or $\underbrace{\boxed{\quad \cancel{<} \quad}}_{75\%}\boxed{x}\underbrace{\boxed{\quad}}_{25\%}$

$T(n) = ?$

$T(1) = c$

$T(n) = cn + T(\frac{3}{4}n)$

Claim: $\cancel{T(n) \le 4cn}$

$T(n) = cn + T(3/4\, n) \le 4 \cdot cn$

Proof by induction: $n = 1 \rightarrow T(1) = c$

Inductive hypothesis: Assume $T(n-1) = c(n-1) + T(\frac{3}{4}(n-1)) \le 4c(n-1)$ is true.

Prove: for n, that $T(n) \le \cancel{4T(3/4\, n)}\ 4cn$

Conjecture: $T(n) \le 4cn$

↙ by substitution of claim

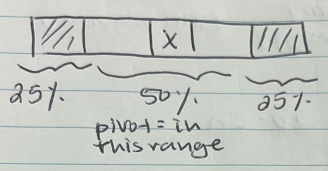
$T(n) = cn + T(3/4\, n) \le cn + 4c(3/4\, n)$

$= cn + 3cn$

$T(n) = 4cn$ □

if we can partition w/ 75% going forward to each next partition → 4cn.

Scenario 2: get rid of 25% from lowest & highest portion.



This decreases runtime from the 75% strategy.

<u>Median of Medians</u>

Get median of medians in groups of 5. Collect the medians of each group of 5. Find the median of the group medians. Sorting is not really performed in this process.

Choose pivot based on the median of medians. Then, pivot will likely end up in the middle 50% of the array.

<u>Questions:</u>

1) Runtime to find median of 5 elements? ~~of medians of groups of 5:~~

$5 \log 5 = c \quad = O(1)$

2) How many groups of 5?

$n/5$

3) How long to find all medians in the groups?

$c(n/5) = O(n)$

4) How long to find m of m?

$T(n/5)$

Runtime for selection algorithm:

$$T(n) = \underbrace{c \cdot n}_{\text{random tasks}} + \underbrace{T(3/4\,n)}_{\text{going into subarray (w/ 75\% strategy)}} + \underbrace{T(n/5)}_{\text{m.o.m (selecting pivot)}}$$

cn: partitioning & housekeeping

Conjecture: $T(n) \leq 20 \cdot cn$

proof by induction: (see before)

prove: $T(n) = c \cdot n + T(n/5) + T(3/4\,n)$

$$T(n) \leq c \cdot n + \cancel{20}^{4} c \left(\frac{n}{\cancel{5}}\right) + \cancel{20}^{5} c \left(\frac{3}{\cancel{4}}\right) n$$

$$T(n) \leq c \cdot n + 4cn + 15cn$$

$$T(n) \leq 20cn$$

Assume n is odd → makes finding median easier.
otherwise, requires using ceil or floor.