374 9/29/22

Lecture 12: Divide and conquer

- Merge sort
- Quick sort
- Quick select
- Binary search ns
- Medians of Medians
- Multiplying polynomials of degree 1
- Multiplying large numbers (Karatsuba's algorithm)

Merge (A, B)

A B - sorted arrays/list

merge ≡ merge the two lists into a single sorted list.

$O(|A|+|B|) = O(n)$ $\quad n = |A|+|B|$.

Merge Sort $(A[1 \ldots n])$:

If $n \le 1$ then return A

$A_L \leftarrow$ Merge Sort $(A[1, \lfloor \frac{n}{2} \rfloor])$

$A_R \leftarrow$ Merge Sort $(A[\lfloor \frac{n}{2} \rfloor + 1, \ldots n])$

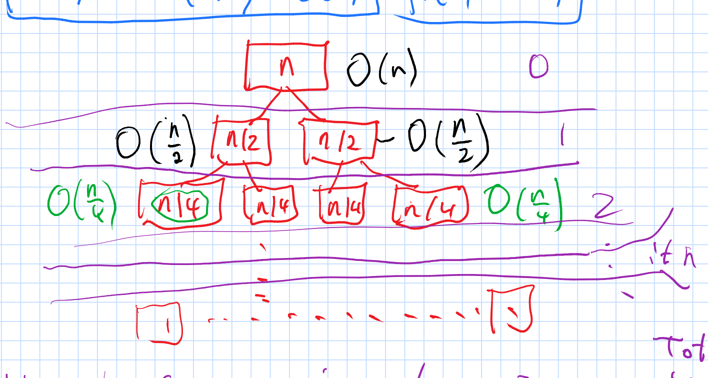
return merge (A_L, A_R)

$T(n) \le O(1) + T(\lfloor \frac{n}{2} \rfloor) + T(\lceil \frac{n}{2} \rceil) + O(n)$

$T(n) \le O(n) + T(\frac{n}{2}) + T(\frac{n}{2})$

$= O(n) + 2T(\frac{n}{2})$

$\boxed{T(n) = 2T(n/2) + O(n)}$ $\boxed{T(1) = O(1)}$



i'th level 2^i nodes

$O(\frac{n}{2^i})$ work at each node

Total contribution of nodes at level i: $O(\frac{n}{2^i}) \cdot 2^i = O(n)$

Height of recursion tree?

$n \to \frac{n}{2} \to \frac{n}{4} \cdots \frac{n}{2^i} \to \cdots 1$

$h = \log_2 n$

Total RT is $h \cdot O(n) = O(n \log n)$.

$H(n) = H(\frac{n}{2}) + 1$ $\quad H(n) = O(\log n)$

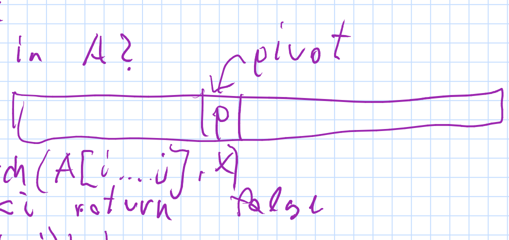
$\boxed{B(n) = B(n/2) + O(1)}$

Binary search

$A[1 \ldots n]$ sorted array

I: value x

Q: Is x in A?



Binary Search $(A[i \ldots j], x)$

If $j < i$ return false

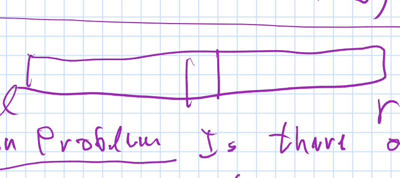
$m \leftarrow \lfloor (i+j)/2 \rfloor$

if $A[m] = x$ return TRUE

if $A[m] > x$ then

return $BS(A[i \ldots, m-1], x)$

return $BS(A[m+1, j])$

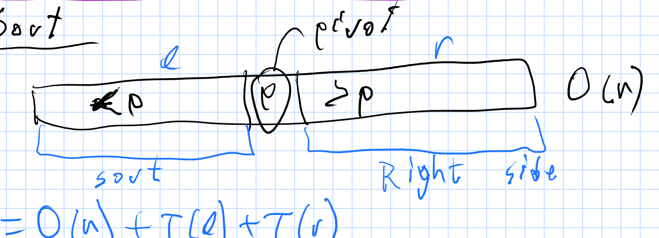


Decision Problem Is there a path between s and t in a graph G with cost at most α?

purple box solving decision problem

Solve search problem?

QuickSort



$Q(n) = O(n) + T(l) + T(r)$

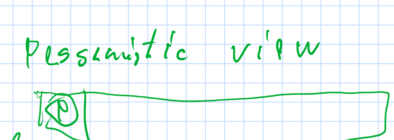
optimistic view

$l = r = \frac{n}{2}$

$O(n) = O(n) + Q(\frac{n}{2}) + Q(\frac{n}{2})$

$= O(n \log n)$

Pessimistic view



$Q(n) = O(n) + Q(0) + Q(n-1)$

$= O(n) + Q(n-1)$

$= O(n) + O(n-1) + Q(n-2)$

$= \ldots$

$= \sum_{i=1}^{n} O(i) = O(n^2)$

Selection

$A[1 \ldots n]$: not sorted

k: number

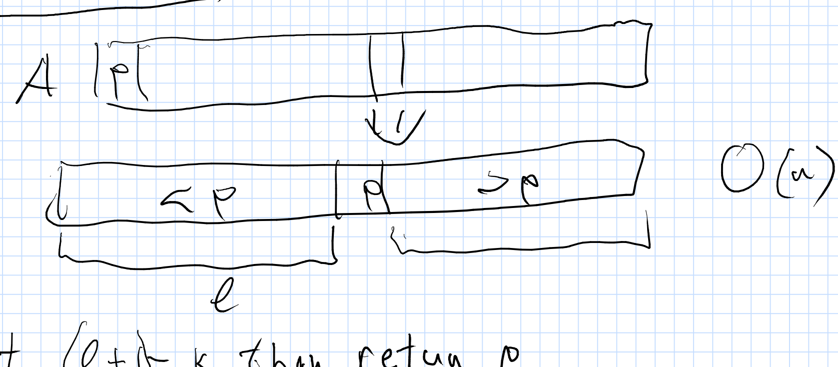
Output the element of rank k in A

output kth smallest number in A

sort ⇒ output kth entry in sorted array

$O(n \log n)$

QuickSelect



If $(l+1) = k$ then return p

If $l \ge k$ then return QuickSelect $(A[1 \ldots l])$

return $QS(A[l+2, \ldots n], k-l-1)$

$Q(n) = O(n) + \max(Q(l), Q(r))$.

$Q(n) = O(n) + \min(Q(l), Q(r))$ optimistic analysis

$= O(n) + Q(\frac{n}{2})$

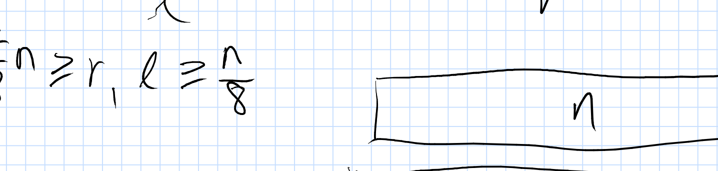
$= O(n) + O(\frac{n}{2}) + Q(\frac{n}{4})$

$= \sum_{i=0}^{\infty} O(\frac{n}{2^i}) = O(n)$

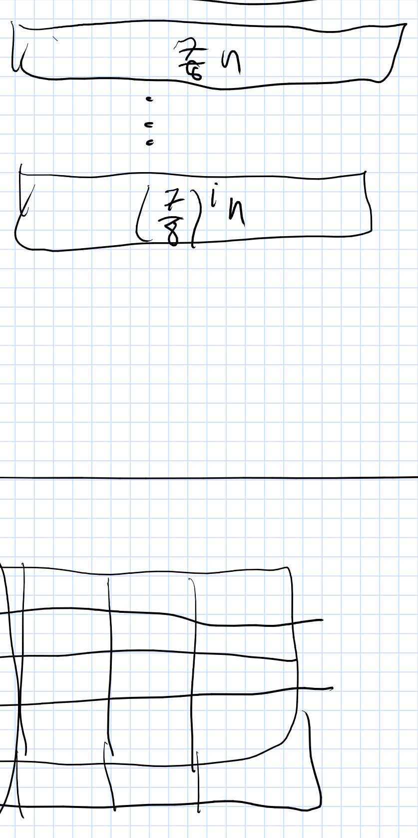
$Q(n) = O(n) + \max(Q(l), Q(r))$ worst case analysis

$= O(n) + Q(n-1)$

$= O(n^2)$



$\frac{7}{8} n \ge r, \; l \ge \frac{n}{8}$



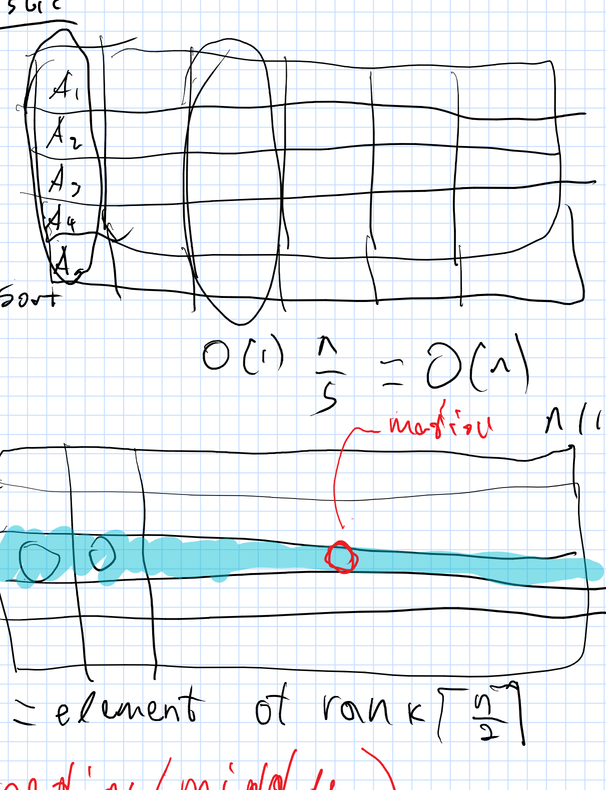
$Q(n) = O(n) + Q(\frac{7}{8} n)$

$Q(n) = \sum_{i=0}^{\infty} O((\frac{7}{8})^i n)$

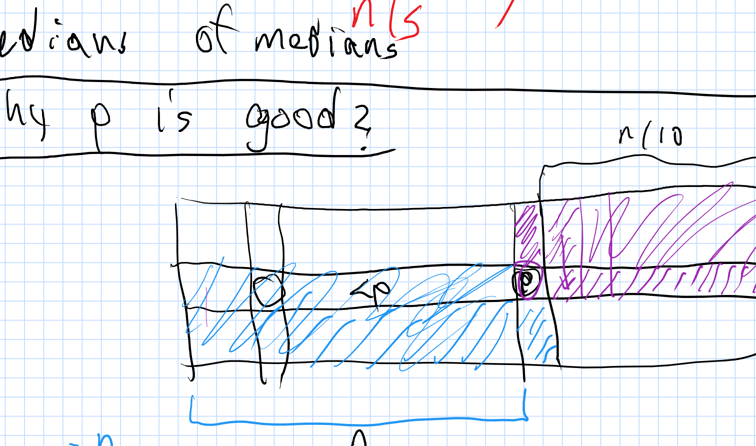
$= O(n \sum_{i=0}^{\infty} (\frac{7}{8})^i)$

$O(n \cdot 1) = O(n)$

Deterministic



$O(1) \; \frac{n}{5} = O(n)$



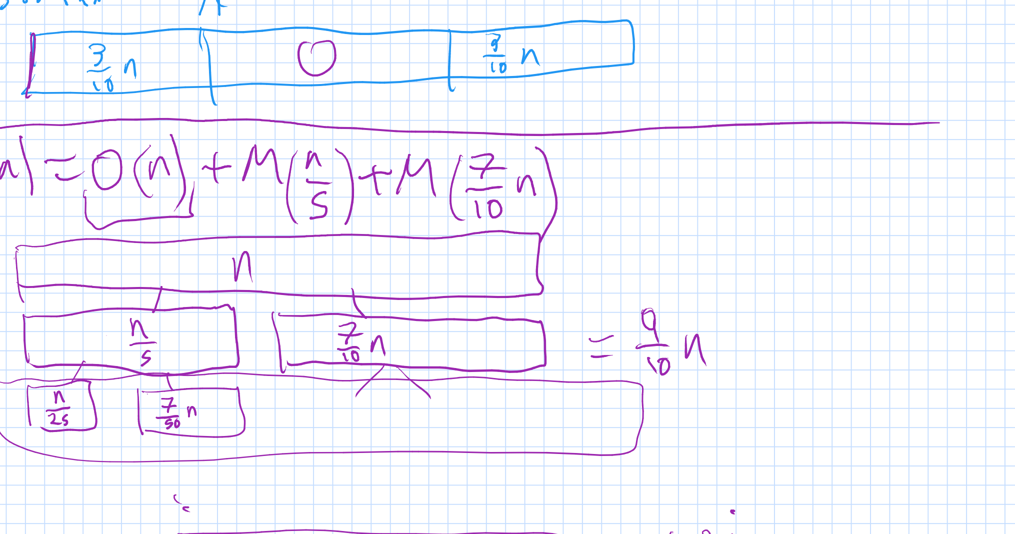
median = element of rank $\lceil \frac{n}{2} \rceil$

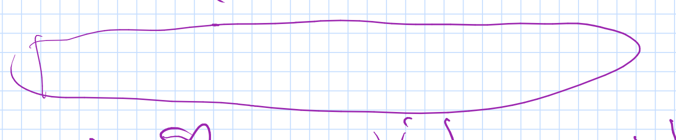
$p \leftarrow$ median(middle elements)

Medians of medians $n/5$

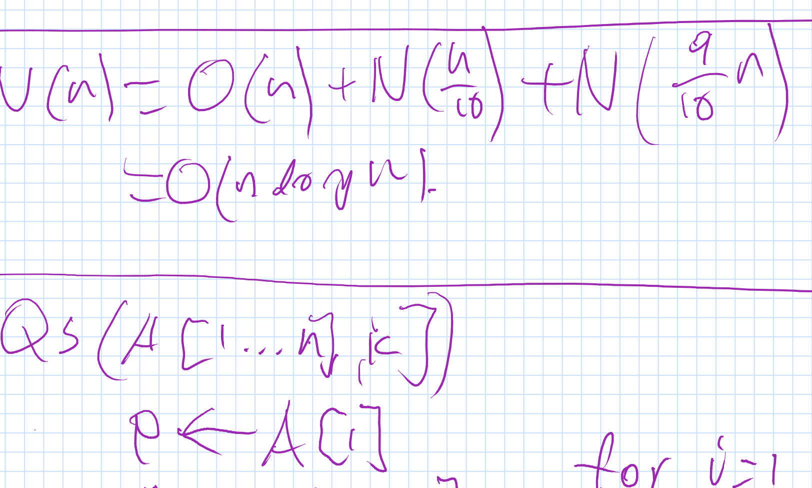
---

why p is good?





$M(n) = O(n) + M(\frac{n}{5}) + M(\frac{7}{10} n)$



$M(n) = \sum_{i=0}^{\infty} O((\frac{9}{10})^i n) = O(n)$

$N(n) = O(n) + N(\frac{n}{10}) + N(\frac{9}{10} n)$

$= O(n \log n)$.

$QS(A[1 \ldots n], k)$

$p \leftarrow A[1]$

$A_L \leftarrow [A < p] =$ for $i = 1$ to n if $A[i] < p$ then A_L.push$(A[i])$

$A_R \leftarrow [A > p]$

$A \leftarrow A_L \,|\, p \,|\, A_R$

If we used 3-tuple instead of 5

$M(n) = O(n) + M(\frac{n}{3}) + M(\frac{2}{3} n) = \Theta(n \log n)$ oy