Lecture 12 (Sorting / Divide & Conquer Algorithms)

Thursday, October 1, 2020 5:00 PM

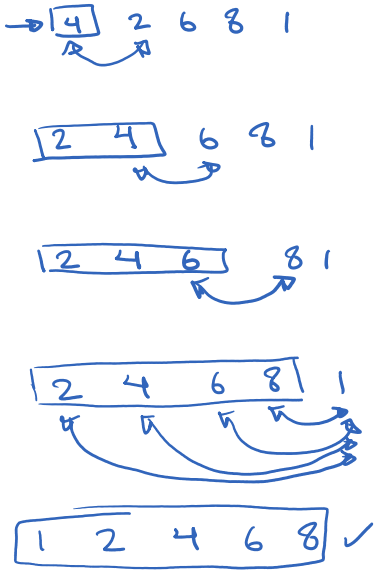
Reminder: * Lab 2 is due this Sunday
* exam 1 is next Thursday

Lab 1: itr=100
2 steps of part B ← Sunday 11:59

Lab 2: .PDF / .java .py
.txt

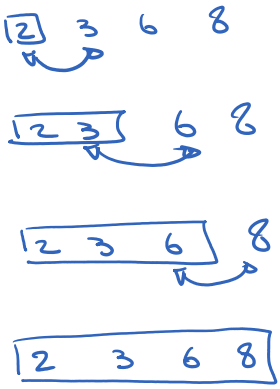
Insertion Sort:

a = [4, 2, 6, 8, 1]



Best-case: $\Omega(n)$

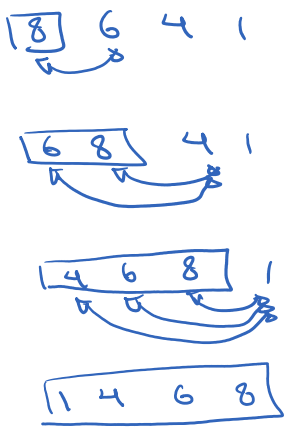
a = [2 3 6 8]



Avg-case is the same

Worst-case: $O(n^2)$

a = [8 6 4 1]

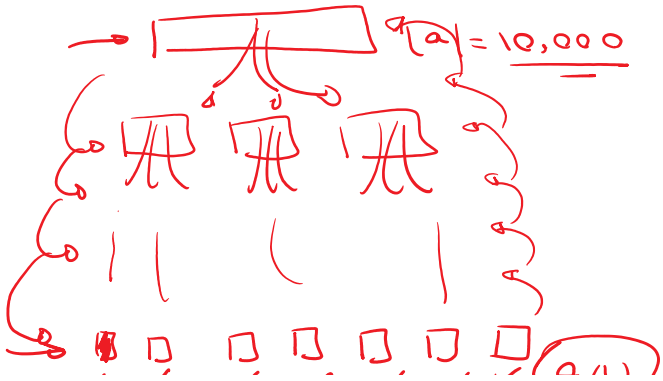


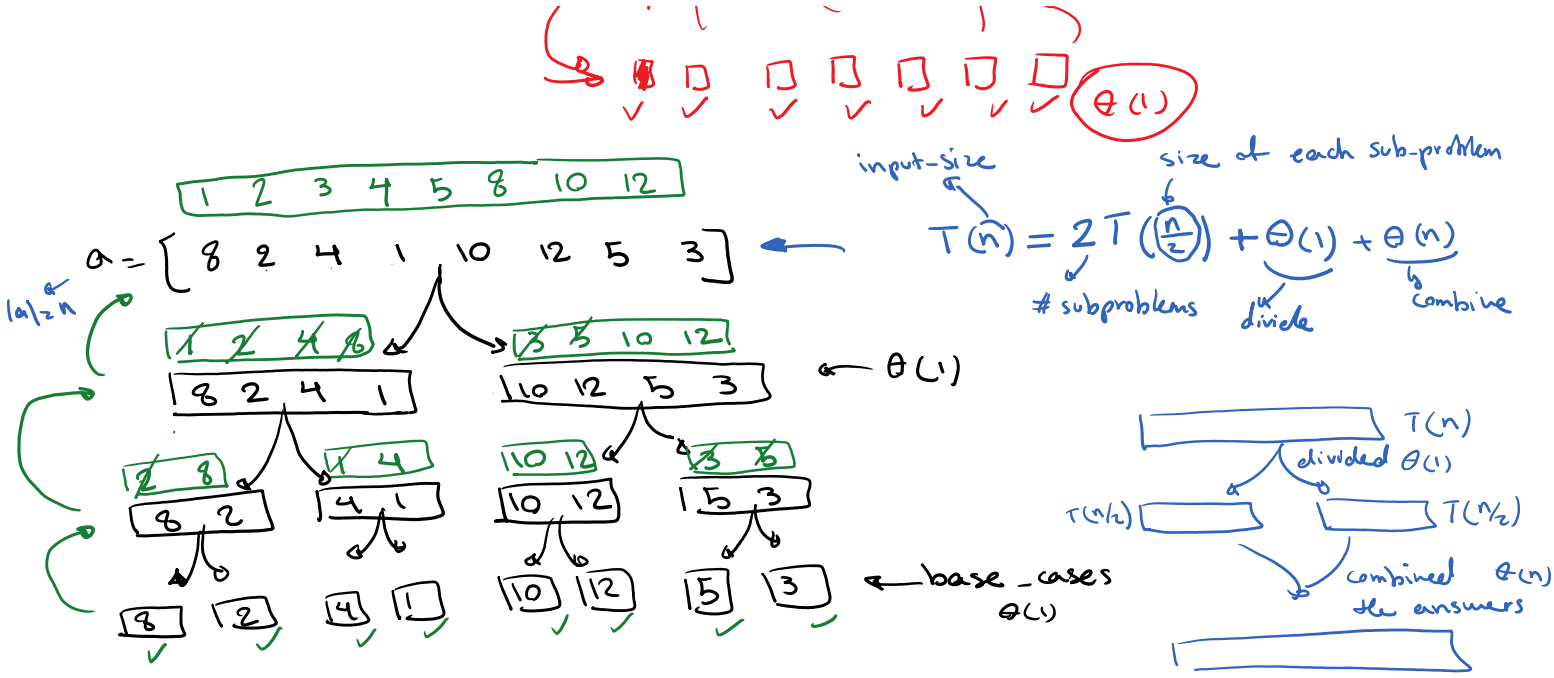
$T(n) = 1 + 2 + 3 + \sim + n-1$

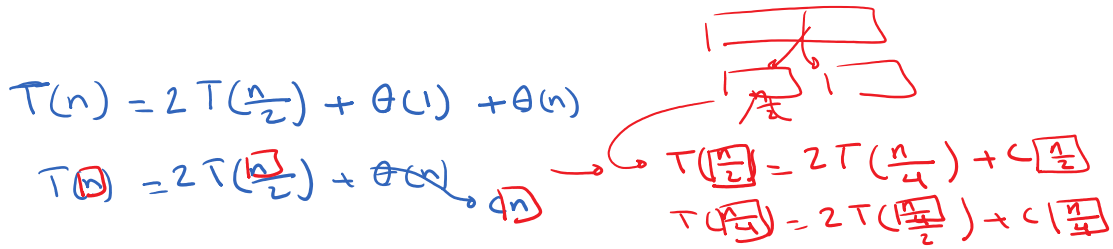
$= \sum_{i=1}^{n-1} i = \frac{(n-1)(n)}{2}$

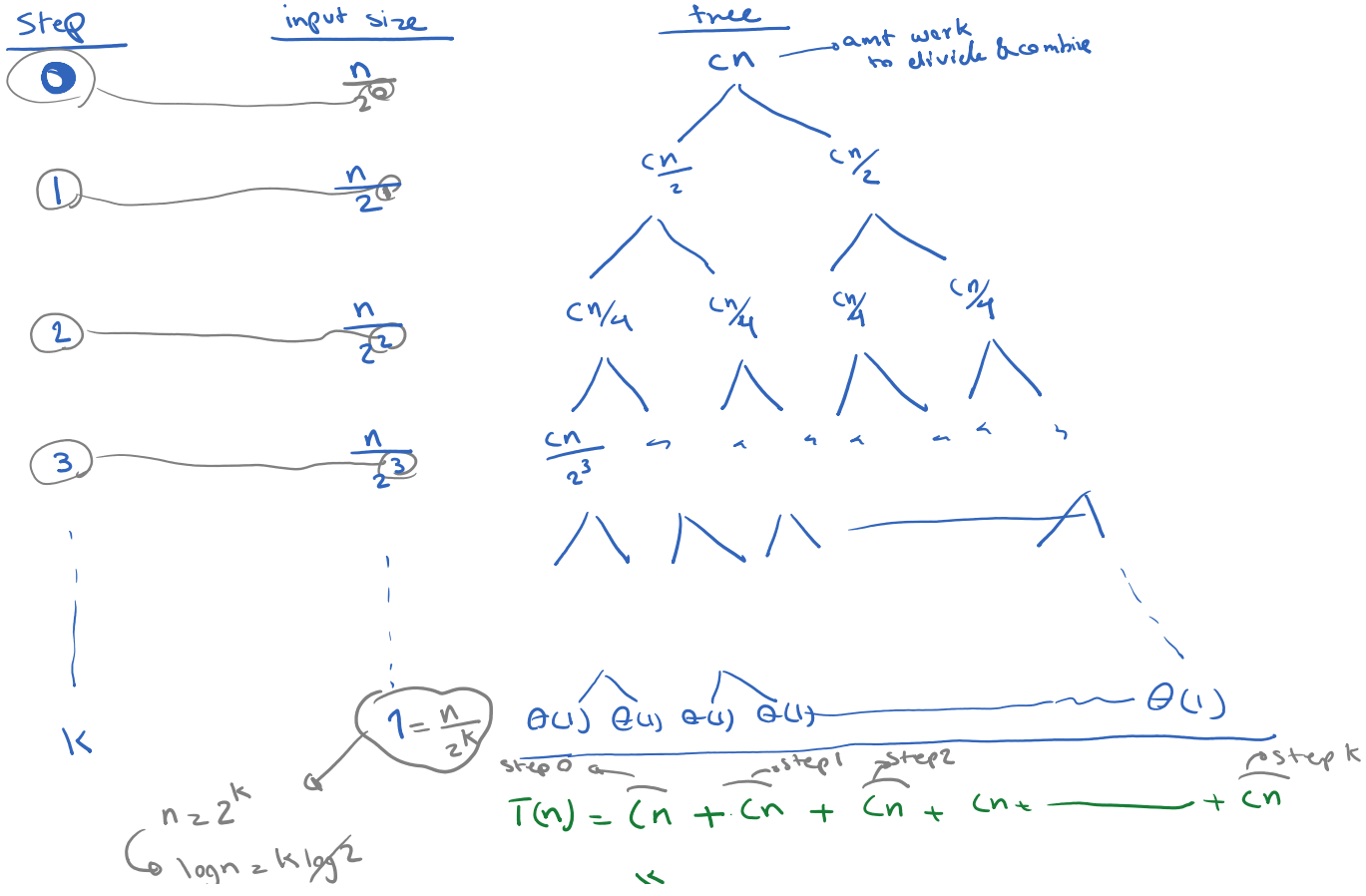
Divide and Conquer Algorithms: ← BS

Merge Sort




θ(1)
input-size
size of each sub-problem
$T(n) = 2T(\frac{n}{2}) + \theta(1) + \theta(n)$
subproblems
divide
combine
1 2 3 4 5 8 10 12
a = [8 2 4 1 10 12 5 3]
|a| = n
base_cases
θ(1)
T(n)
divided θ(1)
T(n/2)
T(n/2)
combined θ(n)
the answers


$T(n) = 2T(\frac{n}{2}) + \theta(1) + \theta(n)$
$T(n) = 2T(\frac{n}{2}) + cn$
$T(\frac{n}{2}) = 2T(\frac{n}{4}) + c\frac{n}{2}$
$T(\frac{n}{4}) = 2T(\frac{n}{8}) + c\frac{n}{4}$


Step
input size
0
$\frac{n}{2^0}$
1
$\frac{n}{2^1}$
2
$\frac{n}{2^2}$
3
$\frac{n}{2^3}$
k
$1 = \frac{n}{2^k}$
$n = 2^k$
$\log n = k \log 2$
tree
cn
amt work to divide & combine
$\frac{cn}{2}$
$\frac{cn}{2}$
$\frac{cn}{4}$
$\frac{cn}{4}$
$\frac{cn}{4}$
$\frac{cn}{4}$
$\frac{cn}{2^3}$
θ(1) θ(1) θ(1) θ(1)
θ(1)
step 0
step 1
step 2
step k
$T(n) = cn + cn + cn + cn + \cdots + cn$

$n = 2^k$

$\log n = k \log 2$

$k = \log n$

$T(n) = cn + cn + cn + \cdots + cn$

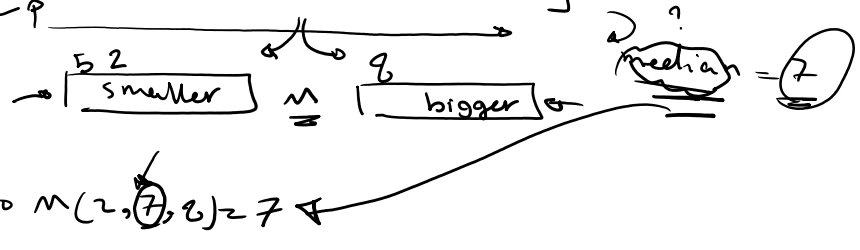
$= \sum_{i=0}^{k} cn = cn(k+1) \sim nk = \Theta(nk)$

$= \Theta(n \log n)$

Quick Sort: $a = [5, 2, 8, 9, 4, 0, 6, 10, 7]$

median

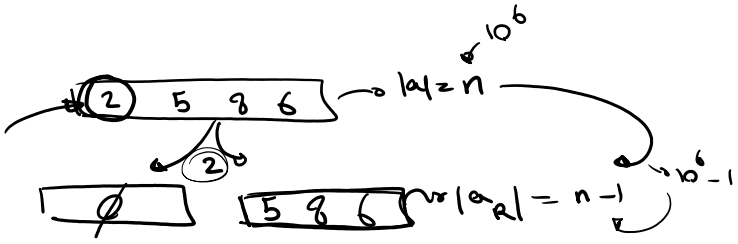
$m(7, 2, 8) \Rightarrow m(2, 7, 8) = 7$



pivot = 1st element / last element / median of three / median of medians / random nr

pivot = 1st element

pivot = 2



Better version → pivot = median of three

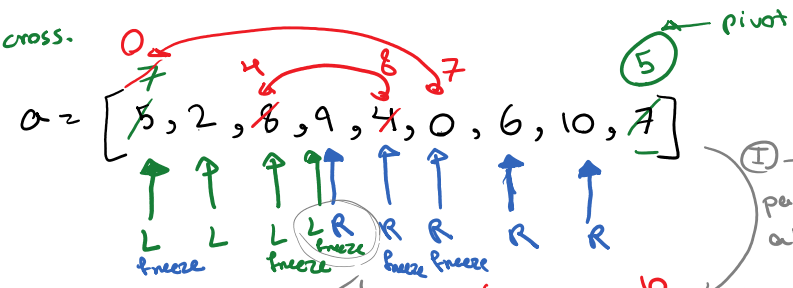
I) pivot = $m(5, 4, 7) = 5$ → $\Theta(1)$

II) Swap (pivot, last element) → $\Theta(1)$

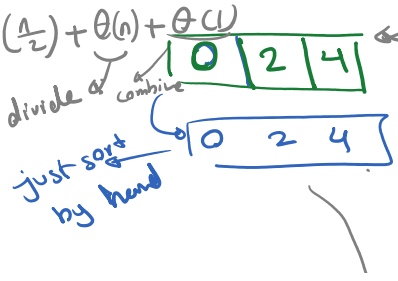
III) Define 2 markers L and R → Divide the array to two → $\Theta(n)$

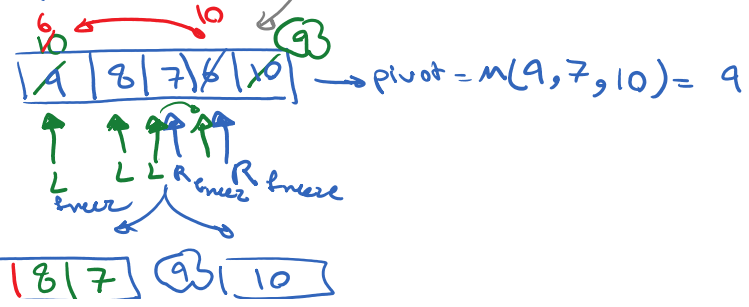
Stop when L & R cross.

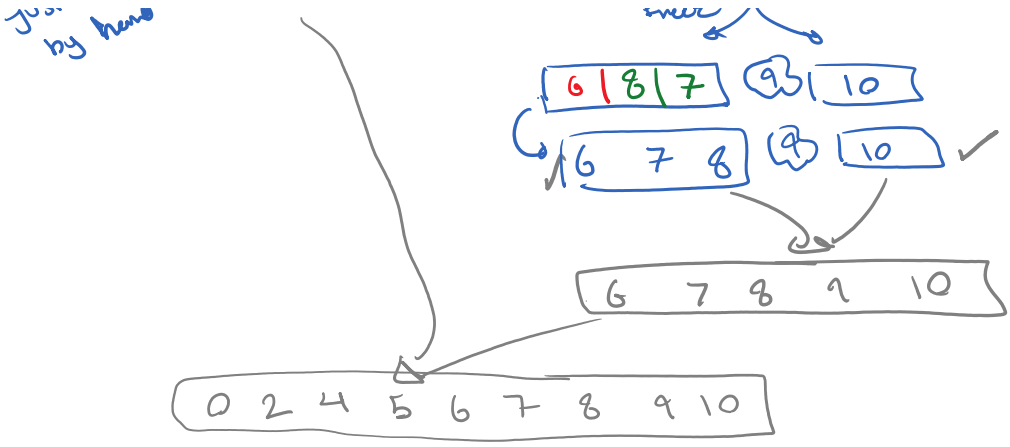
Partitioning Step



$T(n) = 2T(\frac{n}{2}) + \Theta(n) + \Theta(1)$







$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n) + \Theta(1)$$

$\Theta(n)$: divide; $\Theta(1)$: combine

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n) \rightarrow T(n) = O(n \log n)$$

avg-case, best-case

worst-case → HW5