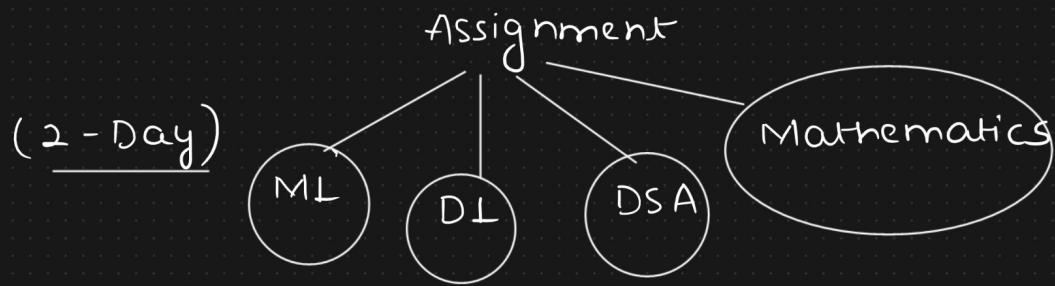


## Divide & Conquer



Sort the array

↳  $n == 1$

↳ small problem

↳ return arr(i)

$n > 1$

↳ big Problem

↳ Divide & Conquer

Approach

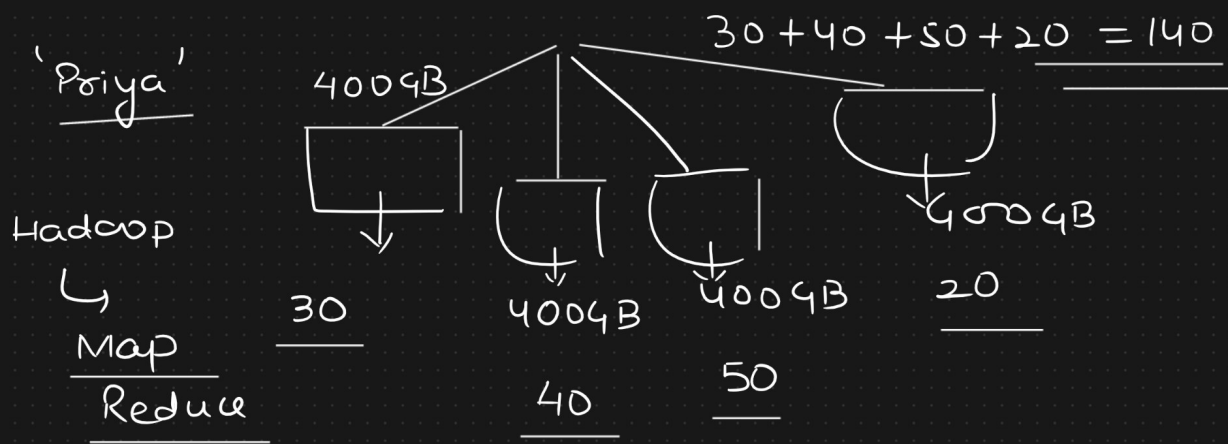
- 1) Divide the problem into various subproblems
- 2) conquer each subproblem
- 3) combine all the solutions (optional)

a) Data Science

↳ b) Data Engineering — **Big Data**

c) Frontend developer

d) none of these



### Pseudocode

divideAndconquer(arr, p, q):

if (small(arr, p, q)).

return solution

else: two parts

Divide ————— m = Divide(arr, p, q)

Recursion

Conquer — {  $b = \underline{\text{divideAndconquer}(arr, p, m)}$

$c = \underline{\text{divideAndconquer}(arr, m+1, q)}$

Combine ————— return combine(b, c)

arr = [75, 45, 95, 50, 60, 67, 29, 32]

0 1 2 3 4 5 6 7

max = 95

min = 29

n = 2

arr(i) < arr(j)

max = arr(j)

min = arr(i)

vice-versa

max = 75 95

min = 75 45 29

O(n)

n = 1

max =

min =

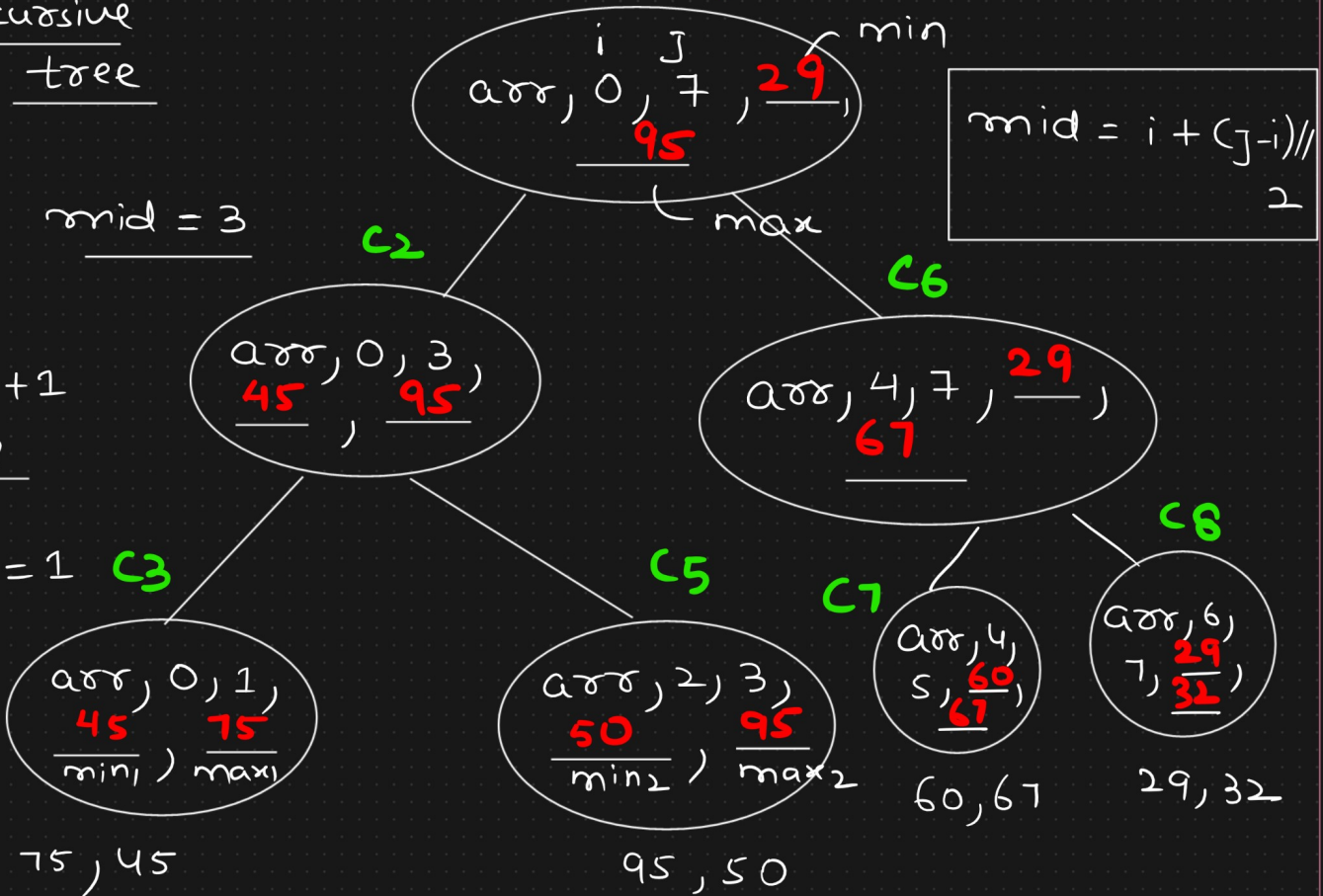
arr(i)

Recursive tree

mid = 3

3 - 0 + 1 = 4

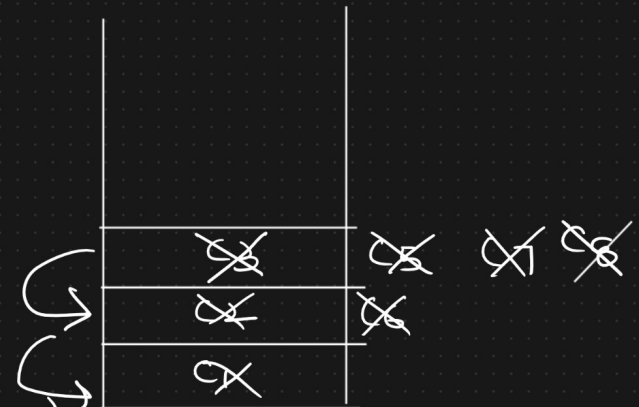
mid = 1



75, 45

single

comparison



(Last In First Out)

LIFO

Stack

Data Structure

# Pseudocode

$T(n)$

$i \rightarrow$  starting index

$J \rightarrow$  ending index

$i, J$

20
----

0

0 1

20	40
----	----

$i = 0$

$J = 1$

Small  
problem  
 $\hookrightarrow c$

findMaxAndMin(arr, i, J):

if  $i == J$ : **single element**

$min = arr(i)$

$max = arr(i)$

elif  $i == J - 1$ : **two element**

if  $arr(i) < arr(J)$ :

$max = arr(J)$

$min = arr(i)$

else

$max = arr(i)$

$min = arr(J)$

else:

$\leftarrow mid = i + (J - i) // 2$

$= \text{findMaxAndMin}(arr, i, mid)$   
 $\hookrightarrow T(n/2)$

$= \text{findMaxAndMin}(arr, mid + 1, J)$   
 $\hookrightarrow T(n/2)$

if  $min_1 < min_2$ :

$min = min_1$

else:

$min = min_2$

$\uparrow c$   
Divide

$\left\{ \begin{array}{l} min_1, max_1 \\ min_2, max_2 \end{array} \right.$

Conquer

$\hookrightarrow 2 T(n/2)$

combine

$\hookrightarrow c$

if  $\max_1 < \max_2$ :  
 $\max = \max_2$   
 else:  
 $\max = \max_1$

return (max, min)

Recurrence Relation:

$$T(n) = \begin{cases} c & n \leq 2 \\ 2T(n/2) + c & n > 2 \end{cases}$$

$$T(n) = 2T(n/2) + c$$

$$a = 2 \quad k = 0$$

$$b = 2 \quad p = 0$$

$$\log_b a = \log_2 2 = 1$$

$$\log_b a > k \rightarrow$$