finding of maxima & minima in an array

arr = [70, 50, 45, 10, 12, 15, 75, 29, 37, 57]
0 1 2 3 4 5 6 7 8 9

Brute force approach

max = ~~70~~ 75

min = ~~50~~ ~~45~~ 10

{ max = 75, min = 10

Best case :- $(n-1).1 = O(n)$

Worst case :- $(n-1).2 = O(n)$

Average case :- $\frac{n-1}{2} \cdot 1 + \frac{n-1}{2} \cdot 2$

$\Rightarrow O(n)$

Divide & Conquer

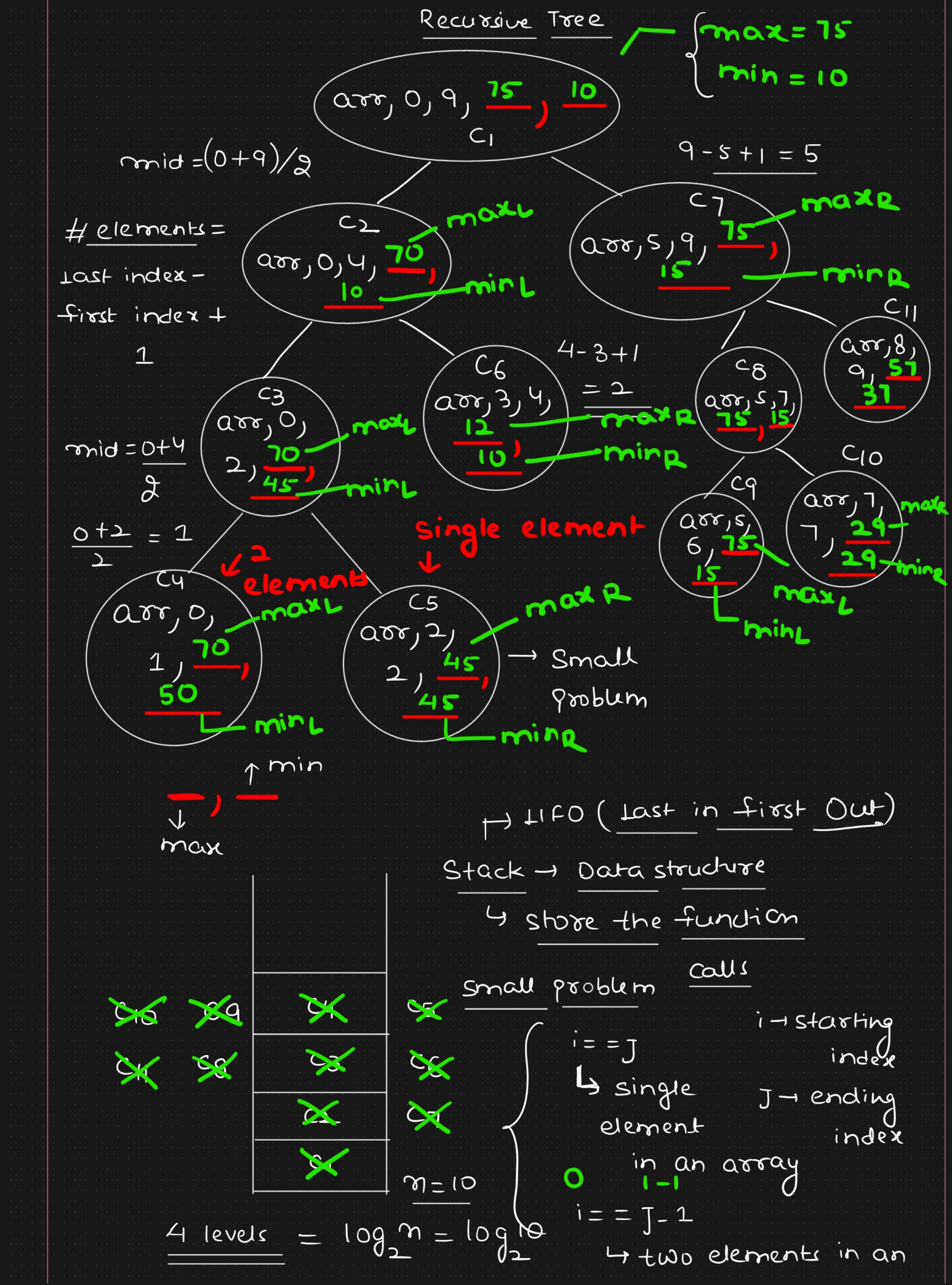
small problem :- $n == 1$ or $n == 2$

$n == 1$ ⇓ return arr { max, min

$n == 2$ ⇓ Single comparison

big problem :- $n > 2$

↳ Divide & conquer


Recursive Tree
max = 75
min = 10
arr, 0, 9, 75, 10
C1
mid = (0+9)/2
9-5+1 = 5
C2
arr, 0, 4, 70, 10
maxL
minL
C7
arr, 5, 9, 75, 15
maxR
minR
elements = last index - first index + 1
C3
arr, 0, 2, 70, 45
C6
arr, 3, 4, 12, 10
4-3+1 = 2
C8
arr, 5, 7, 75, 15
C11
arr, 8, 9, 57, 31
mid = 0+4/2
0+2/2 = 1
C4
arr, 0, 1, 70, 50
2 elements
C5
arr, 2, 2, 45, 45
Single element
C9
arr, 5, 6, 75, 15
C10
arr, 7, 7, 29, 29
min
max
Small problem
LIFO (Last in first Out)
Stack → Data structure
↳ store the function calls
i → starting index
J → ending index
i == J
↳ single element in an array
i == J - 1
↳ two elements in an
n = 10
4 levels = log2 n = log10 / log2

Stack space = O(log n)

array

Psuedocode

i → starting index

J → ending index

T(n) ⇒ findminAndmax(arr, i, J):

Small problem ↓↓ C

if i == J: → Single element in an array

min = arr(i)

max = arr(i)

elif i == J-1: → two elements in an array

if arr(i) < arr(J):

min = arr(i)

max = arr(J)

else:

min = arr(J)

max = arr(i)

else: ⌐ Divide — C

mid = i + (J-i)/2

conquer

max_L, min_L = findmaxAndmin(arr, i, mid) — T(n/2)

max_R, min_R = findmaxAndmin(arr, mid+1, J) — T(n/2)

combine → C

if $max_L < max_R$:

max = max_R

else:

max = max_L

```
if min_L < min_R :
    min = min_L
else :
    min = min_R
```

return (max, min)

Recurrence Relation

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

⟶ Substitution Method

↳ $O(n)$

Master's Theorem

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

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

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

$$\log_b^a > k$$

↳ $\theta\left(n^{\log_b^a}\right)$

$\Rightarrow \theta(n^1)$

$\Rightarrow \theta(n)$