

2.5 years

Keshav Sekhsaria  
IIT BHU (Varanasi) MnC

- 1 year gfg
- 1.5 year scalon

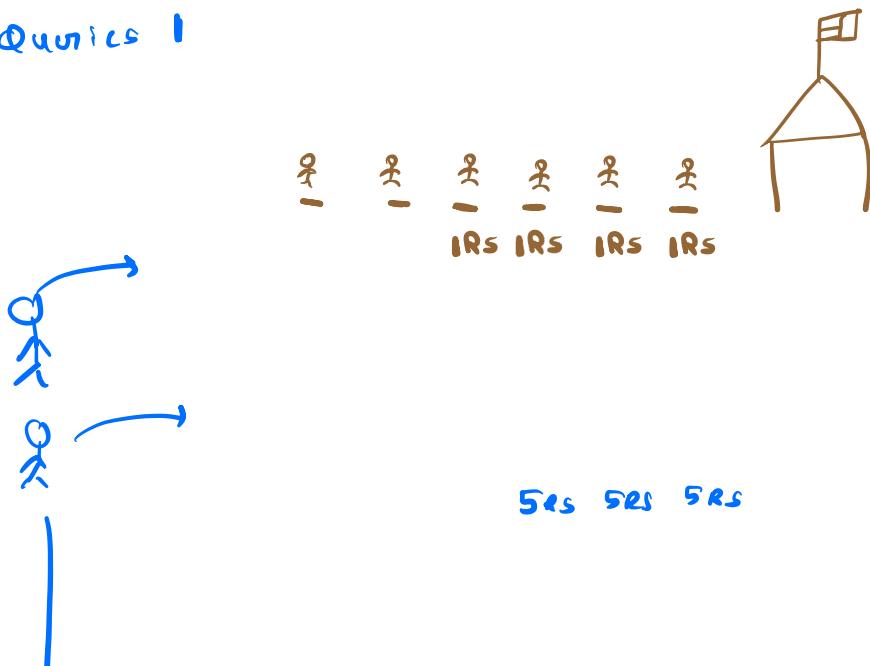
WorldQuant Quant Researcher

Google SWE (BLR)

## Today contents

- ✓ Zoro Quenics I (Beggar's Problem)
- ✓ Zoro Quenics II
- ✓ Rain water Trapping
- ✓ Max Subarray sum (Kadane's Algo)

## I) Zoro Quenics I



Given  $\Delta$  quaries

↓  
start ind  
val

(Start ind) start, and increase ind to  $N-1$   
by val.

Return Find status of each beggar.

Q      arr = [ 0 0 0 0 0 0 ]  
              0 1 2 3 4 5  
              H +4 +4

+3 +3 +3 +3 +3

-2 -2

ind      val  
3      4  
1      3  
4      -2

[ 0, 3, 3, 7, 5 5 ]  
arrs

BF.

For each query : iterate from ind  
to  $N-1$  & update the values of the arr.

return arr[]

TC :  $O(QN)$

SC :  $O(1)$

Idea2 arr = 

$a_0$	$a_1$	$a_2$	$a_3$	$a_4$
-------	-------	-------	-------	-------

$$\begin{array}{cccccc} \text{Prefix} & = & a_0 & a_0 & a_0 & a_0 \\ & & + & + & + & + \\ & & a_1 & a_1 & a_1 & a_1 \\ & & + & + & + & \\ & & a_2 & a_2 & a_2 & \\ & & + & + & & \\ & & a_3 & a_3 & & \\ & & + & & & \\ & & a_4 & & & \\ \hline & & 0, 0, 0, 0, 0 & & & \end{array}$$

$\alpha = 3$

$$\text{arr} = [ \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 4 & -2 \end{matrix} ]$$

$\alpha = 3$   $\downarrow$

$$pr = [0, 3, 3, 7, 5, 5]$$

Ind	Val
3	4
1	3
4	-2

$\alpha = 4$

$$\text{arr} = [ \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 4 & -2 \\ 5 \end{matrix} ]$$

Ind	Val
3	4
1	3
4	-2
3	1

$$PF[] = [0, 3, 3, 8, 6, 6]$$

$\text{arr}[] = \{0\}$

for ( $i=0$ ;  $i < Q$ ;  $i++$ )

{  
     $\text{ind}, \text{val}$   
     $\text{arr}[\text{ind}] += \text{val}$

Create PF sum[] from arr  
return PFsum

TC:  $O(Q+N)$

SC:  $O(N) \rightarrow O(1)$

Q. Initially  $\text{arr}[0]$  all elements are 2010.

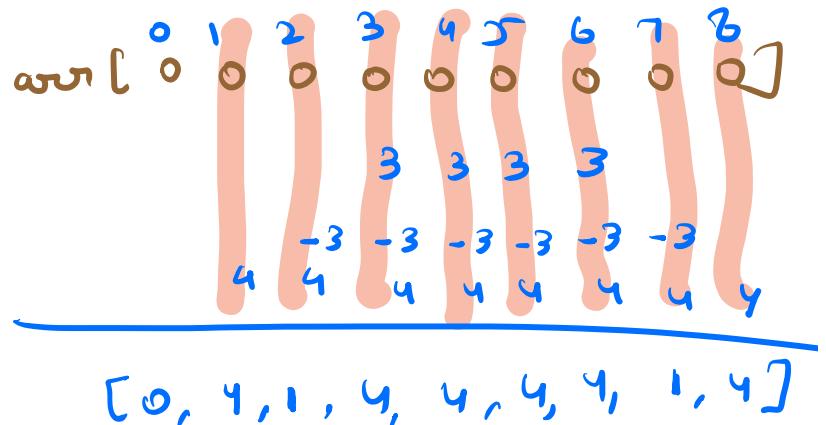
You are given  $Q$  queries,  $[s, e, val]$ .

Increment values of elements from  $s$  to  $e$  by  $val$ .

Find final state of the  $\text{arr}[]$

$$\varrho = \beta$$

s	e	val
3	6	+ 3
2	7	- 3
1	8	4



`arr[] = arr[ 0 0 0 0 0 0 0 0 ]  
          3           -3`

$$\begin{array}{r} 3 \\ + 6 \\ \hline \end{array}$$

$$PF = [0, 0, 0, 3, 3, 3, 3, 0, 0]$$

արտօնություն

```
for ( i=0 ; i<Q ; i++ )
```

S, e, val

,

arr[S] += val

if (e < N - 1) arr[e + 1] -= val

Create PF sum[] from any

return  $\rho_{\text{sum}}$

$$TC: O(Q + N)$$

$$SC: O(N) \xrightarrow{\downarrow} O(1)$$

$$Q = 3$$

$$\text{arr} [ \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 4 & -3 & 3 & & & & & -3 & 3 \end{matrix} ]$$

s	e	val
3	6	+3
2	7	-3
1	8	4

$$PF = [0, 4, 1, 4, 4, 4, 4, 1, 4]$$

Q. Given arr[].

Find lmax[]

$lmax[i] = \max$  of all elements from  $0 \rightarrow i$

$$\text{arr}[] = [1, -6, 3, 8, 4, 5, 2]$$

$$lmax[] = [1, 1, 3, 8, 8, 8, 8]$$

$\text{imax}[0] = \text{arr}[0]$

for ( $i=1$ ;  $i < N$ ;  $i++$ )

{      $\text{imax}[i] = \max(\text{imax}[i-1], \text{arr}[i])$

return  $\text{imax}[N]$

Q. Given  $\text{arr}[ ]$ .

Find  $\gamma_{\text{max}}[ ]$

$\gamma_{\text{max}}[i] = \max$  of all elements from  $i$  to  $N-1$

→ TODO

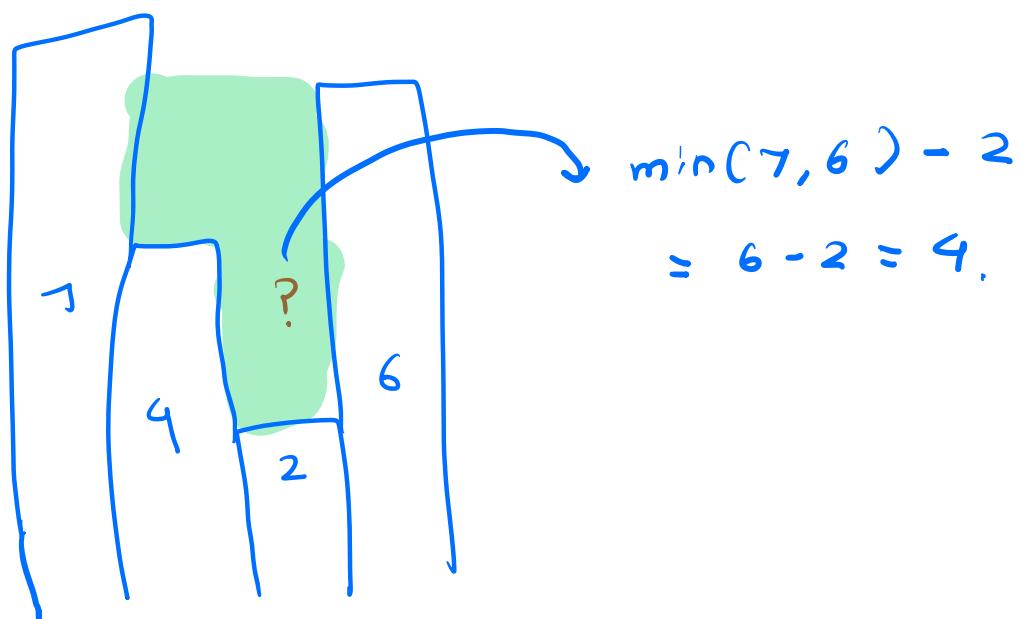
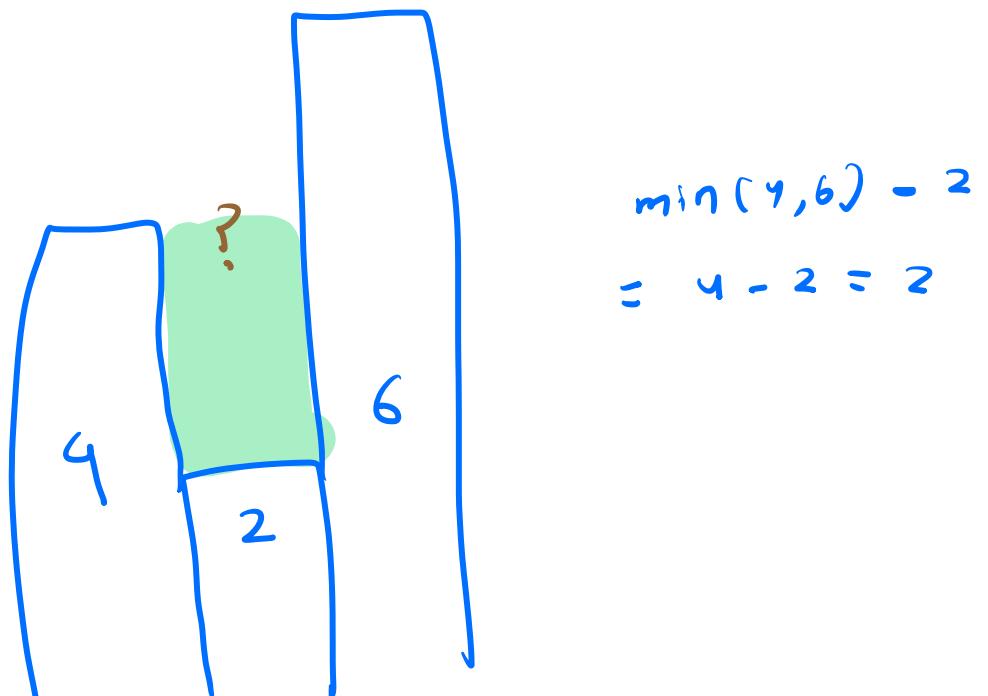
Q.



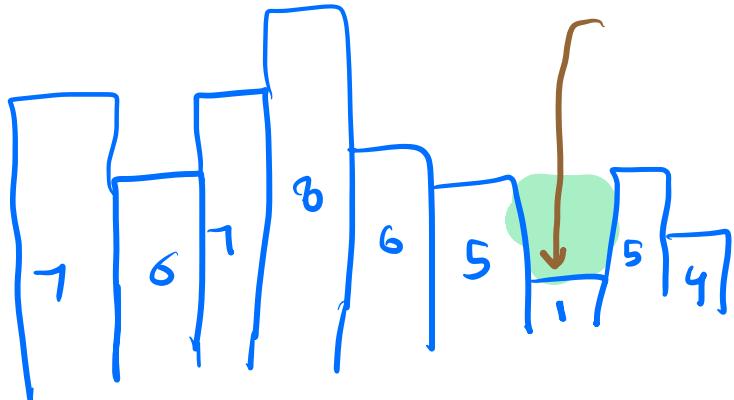
Given N Buildings (heights), Find total water stored between these building.



Total water = sum of water on each building.



for each building =  $\min(d_{\max}, x_{\max})$   
 $- \text{area of } i$

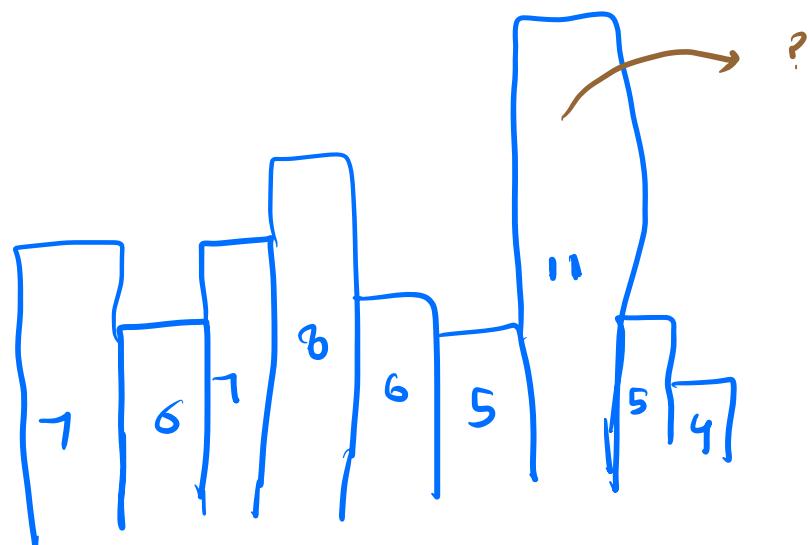


$$l_{\max} = 8$$

$$r_{\max} = 5$$

$$= \min(8, 5) - 1$$

$$= 5 - 1 = 4$$



$$l_{\max} = 11$$

$$r_{\max} = 11$$

$$= \min(11, 11) - 11$$

$$= 0$$

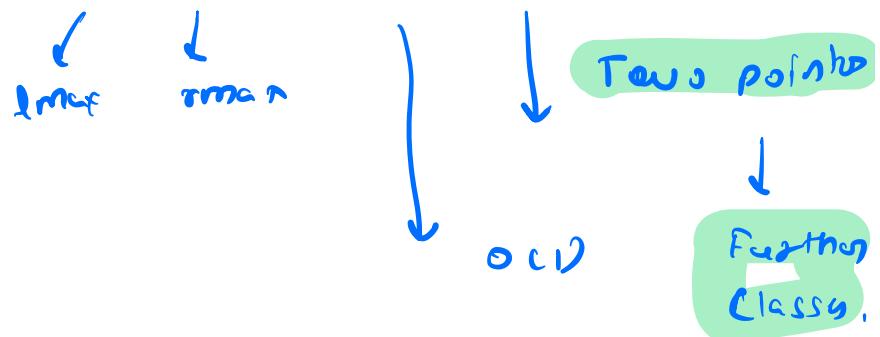
```

ans = 0
for (i=0; i<n; i++)
{
    curr = min(lmax[i], rmax[i]) - arr[i]
    ans += curr
}
return ans

```

$$TC: O(N+N+N) : O(N)$$

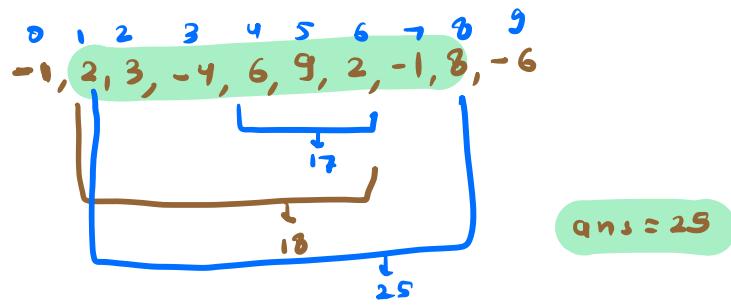
$$SC: O(N+N) : O(N)$$



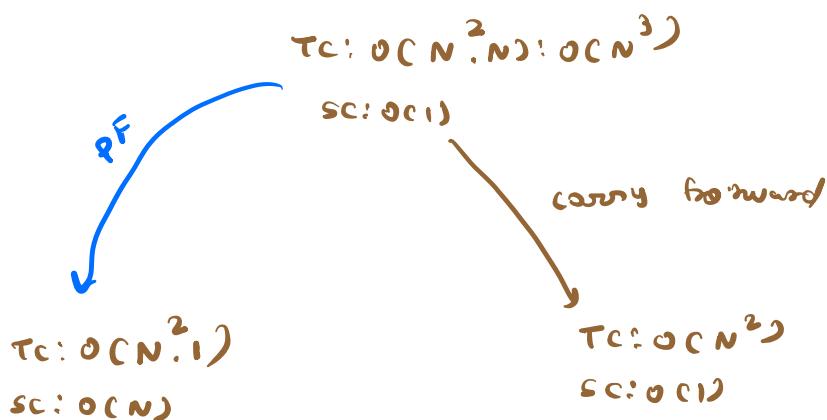
$$b-a+1$$

$$29-20+1 = 10$$

Q. Find maximum subarray sum.



B.F. Consider each subarray. Cal sum  
update max



✓ # All elements are positive

$$3, 5, 9, 7 = \text{sum of arr}$$

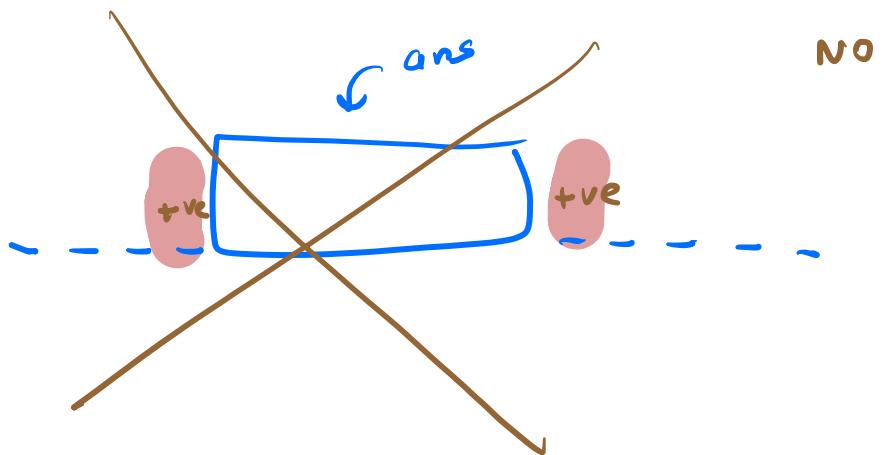
✓ # All elements are negative

$$-3, -7, -11, \boxed{-1}, -6 \quad \text{ans} = -1$$

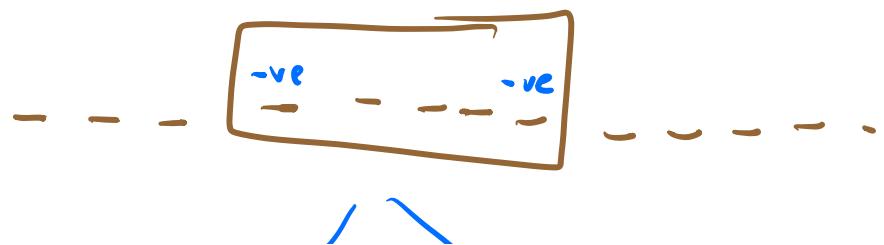
✓ # All value positive ( Except boundary are negative )

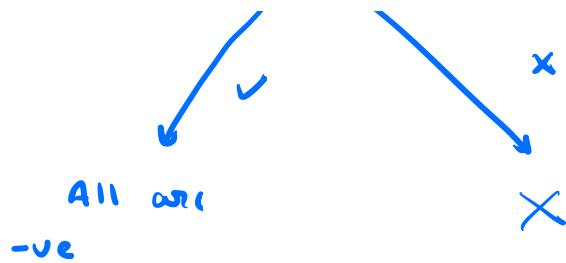
- 3, 7, 11, 3, 5, -8 ans.

Q. Can the no be there outside boundary?



Q. Can -ve no be there outside boundary?





5, 6, 7, -3, 2, -10, -12, 8, 12, 21, -4, 7 -5

$\text{curr sum} = 0$	5	11	18	15	17	7	15	8	20	41	37	44	39
$\text{max ans} = 0$	5	11	18	18	18	18	18	18	20	41	41	44	44



### Kadane's Algo

ans  
INT\_MIN

sum = 0, ans = -∞ / arr[0]

for (i=0; i < N; i++)

```
{
    sum = sum + arr[i]
    ans = max(ans, sum)
    if (sum < 0) sum = 0
}
```

return ans

Tc: O(N)  
Sc: O(1)

	-3	-4	-7	
s=0	-30	-40	-70	
a=0	-3	-3	-3	-3

X ————— X —————

## 1. Zero Queries I

for each query  
 $\{$  arr[ind] += val

3 6 +3



✓PF sum

3 END +3

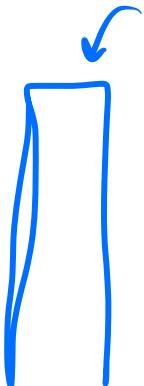
## 2. Zero Queries II

2 END -3

for each query  
 $\{$  arr[ind] += val  
arr [end+1] -= val

✓PF sum

### 3. Rain water

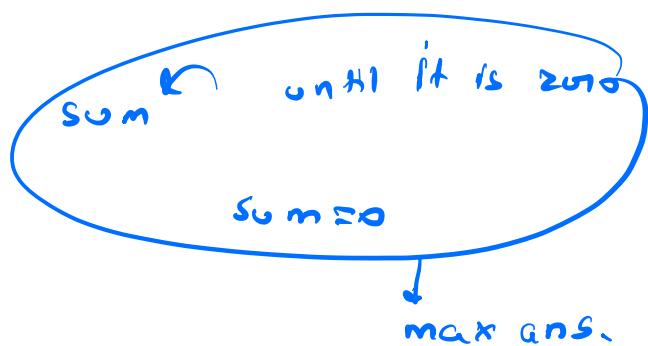


$$\text{ans} = \sum_{j=0}^{n-1} \text{water}[j] \leq \min(l_{\max}, r_{\max}) - \text{arr}[j]$$

### 4. Max Subarray sum

$\downarrow$   
 $T.C: O(N^2)$

Kadane's algo.

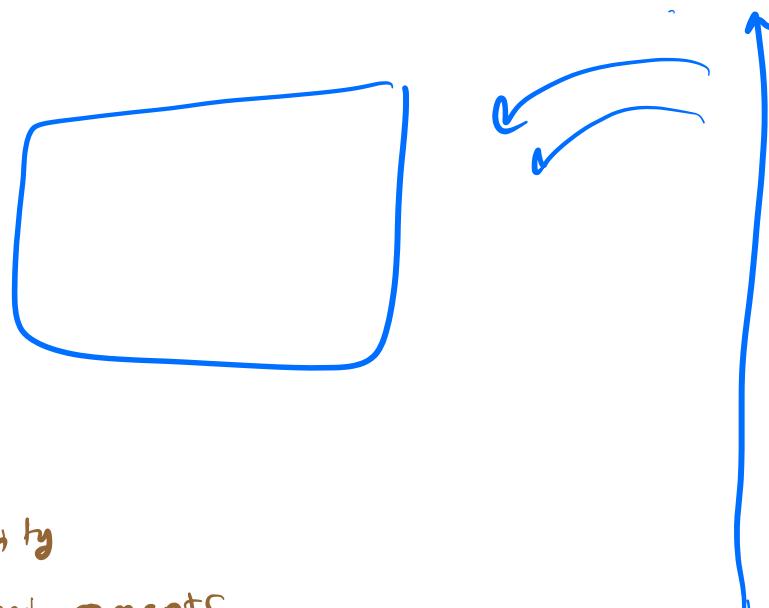


Code

sum = 0, ans = -∞ / arr[0]

for (i = 0; i < n; i++)

{  
    sum = sum + arr[i]  
  
    if (sum > ans)  
        e → s →  
        len++  
  
    if (sum < 0)   sum = 0  
                   s = i+1  
                   l = 0  
  
return ans

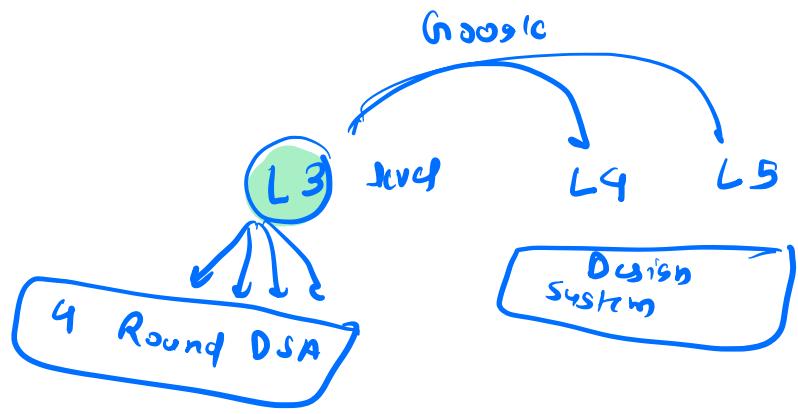


Priority

\* 1. Assignments

\* 2. Homework

\* 3. Leetcode



0 1 2 3 4 5 6  
 0 0 0 0 0 0 -3  
 s e 3  
 1 . 5 ~~3~~  
 = 0, 3, 3, 3, 3, 3, 0

