

One Dimensional Array

TABLE OF CONTENTS

1. Max Subarray Sum
2. Prefix Sum
3. Rain Water Trapped



3. merging intervals



< Question > : Given arr[N]. Find the maximum subarray sum out of all subarrays.

$$1 \leq N \leq 10^5$$

Example 1 : $\text{arr[]} \rightarrow [-3, \overbrace{2, 4, -1, 3}^8, -4, 3]$

0 1 2 3 4 5 6

Example 2 : $\text{arr[]} \rightarrow [\overbrace{4, +5, +2, +1, 6}^8]$ 18 ans

0 +1 +2 +3 +4

GGF GF F Me S 18

Example 3 : $\text{arr[]} \rightarrow [\overbrace{-4, -3, -6, -9, -2}^8]$

.

GGF GF F Me S

-4 -3 -6 -9 -2

Idea -1

-2 ans

$$[1, -2, 3, 4]$$

$$N = 4$$

$$N * (N+1)$$

$$\frac{2}{2}$$

$$= \cancel{\frac{1}{2}} \times 5 = 10$$

1>	1
1, -2	-1
1, -2, 3	2
1, -2, 3, 4	6
-2	-2
-2, 3	1
-2, 3, 4	5
3	3
3 4	7
4	4

Brute force idea :

\Rightarrow 2 loops to generate s and e



$$TC = \underline{O(N^2)}$$

prefix sum

$$SC = O(N)$$



Observations

Case 1

When all elements are positive

2	4	5	1	3
---	---	---	---	---

maximum

$$\text{sum} = \text{sub_sum}$$

Case 2

When all elements are negative

-7	-3	-11	-8	-15
$\overline{-7}$	$\overline{-3}$	$\overline{-11}$	$\overline{-8}$	$\overline{-15}$

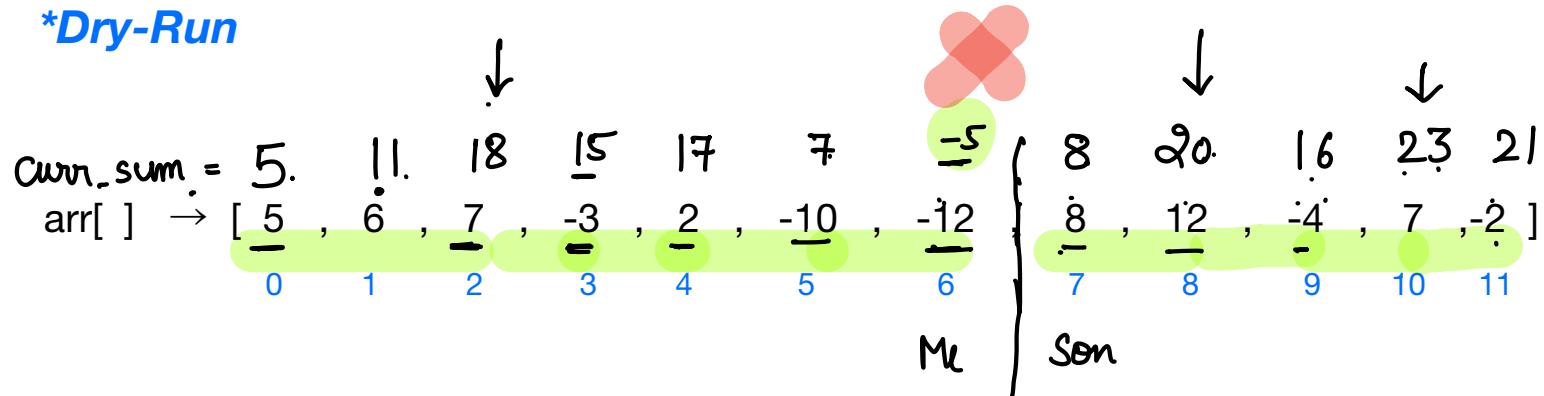
-3 ans

Case 3

	+ve	
--	-----	--



*Dry-Run

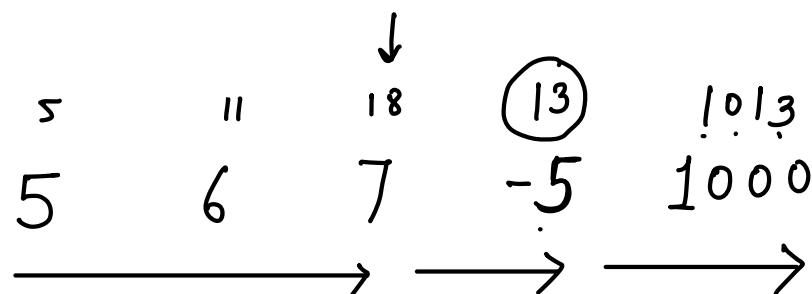


Take = +ve inheritance

or

$$\text{Leave} = -ve$$

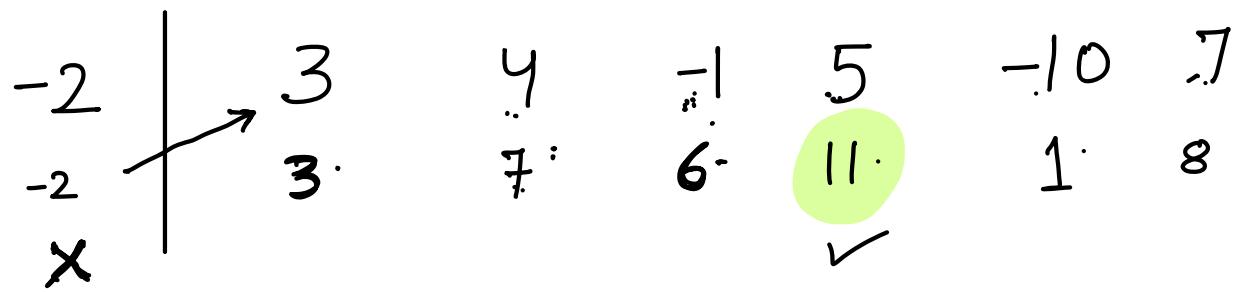
23



$$\begin{array}{r}
 5 \\
 5 \\
 \hline
 11 \\
 6 \\
 \hline
 18. \\
 7 \\
 \hline
 -20 \\
 \hline
 . \\
 X \\
 \hline
 1000 \\
 \hline
 998
 \end{array}$$

+ve

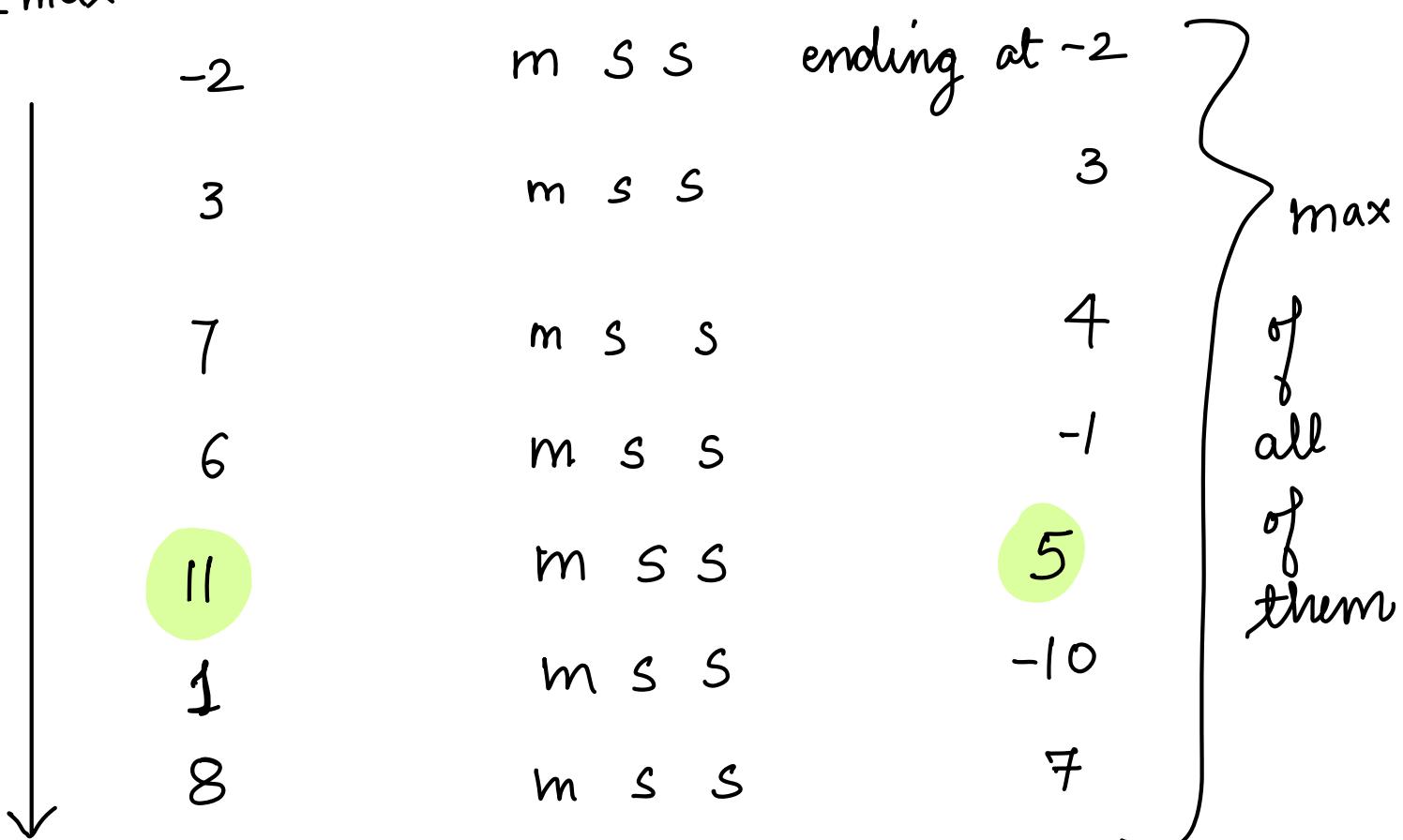
$$\begin{array}{r}
 & 11 & 18 & 13 & -987 \\
 5 & 6 & 7 & -5 & -1000 \\
 & & & & \times \\
 \hline
 & & & & \left| \begin{array}{c} 1 \\ 1 \end{array} \right. \\
 & & & & \swarrow \quad \searrow
 \end{array}$$



$$\text{ans} = 11$$

max subarray sum - out of subarray sums,
find max.

Ov-max



$$m \ s \ s =$$

```
int m_Sum (int[] arr ) {  
    int N = arr.length  
  
    curr_max = arr[0]  
    ov_max = arr[0]  
  
    for ( i=1      i < N      i++) {  
        if (curr_max > 0)  
            curr_max = curr_max + arr[i]  
        else  
            curr_max = arr[i]  
  
        if (curr_max > ov_max)  
            ov_max = curr_max  
  
    }  
  
    return ov_max
```

{

i=1
 -2 | 3 4 -1 5 -10 7
 c-m -2 | 3 7 6 11 1 8
 o-v -2 | 3 7 7 11 11 11
 11

```

int m_ssum(int[] arr) {
    int N = arr.length;
    curr_max = arr[0];
    ov_max = arr[0];
    for (i=1; i < N; i++) {
        if (curr_max > 0)
            curr_max = curr_max + arr[i];
        else
            curr_max = arr[i];
        if (curr_max > ov_max)
            ov_max = curr_max;
    }
    return ov_max;
}
  
```

-4 | -11 -5 -2 -3
 c-m -4 | -11 -5 -2 -3
 o-m -4 | -4 -4 -2 -2

```

int m_ssum(int[] arr) {
    int N = arr.length;
    curr_max = arr[0];
    ov_max = arr[0];
    for (i=1; i < N; i++) {
        if (curr_max > 0)
            curr_max = curr_max + arr[i];
        else
            curr_max = arr[i];
        if (curr_max > ov_max)
            ov_max = curr_max;
    }
    return ov_max;
}
  
```

-2 ans

max Subarray Sum ✓

print the subarray, \rightarrow MSS





< Question > : Initially all elements of an arr[N] are 0. Then you are given Q queries. Every query contains i-idx and value. Increment elements from ith.idx to last idx by value. Return final state of arr[].

arr[] → [0 0 0 0 0 0 0]
 0 1 2 3 4 5 6 :
 . . .
 , +4 +4 +4 +4
) +3 +3 +3 +3 +3 +3
 +3 -2 -2 -2
 0 3 3 7 5 5 5 ← arr



BF Idea

2 loops

for every query \longrightarrow

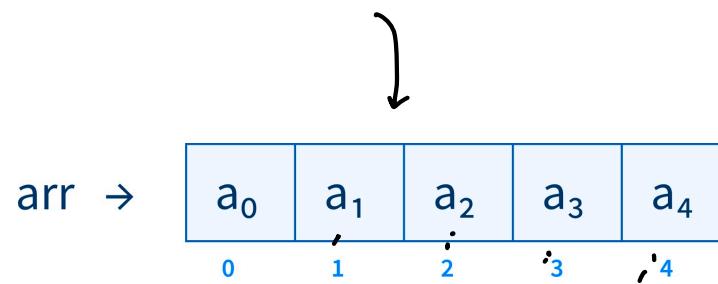
$$i - N - 1$$

$\text{arr}[i] = \text{arr}[i] + \text{val}$

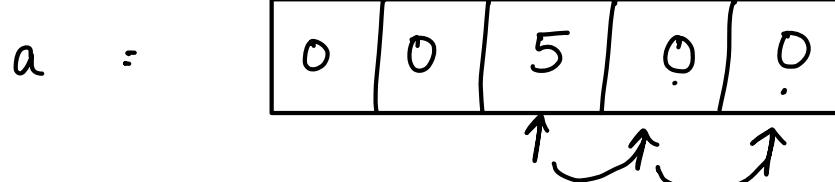
$$\begin{array}{cccccc}
 & 0 & 1 & 2 & 3 & 4 \\
 0 & 0 & 0 & 0 & 0 & 0 \\
 \\
 |, 3 & & +3 & +3 & +3 & +3 \\
 \\
 0, 2 & +2 & +2 & +2 & +2 & +2 \\
 \\
 4, ! & \downarrow & \downarrow & \downarrow & \downarrow & +1 \\
 \\
 \hline
 2 & 5 & 5 & 5 & 6 & \text{ans} \\
 \hline
 \end{array}$$



Observation

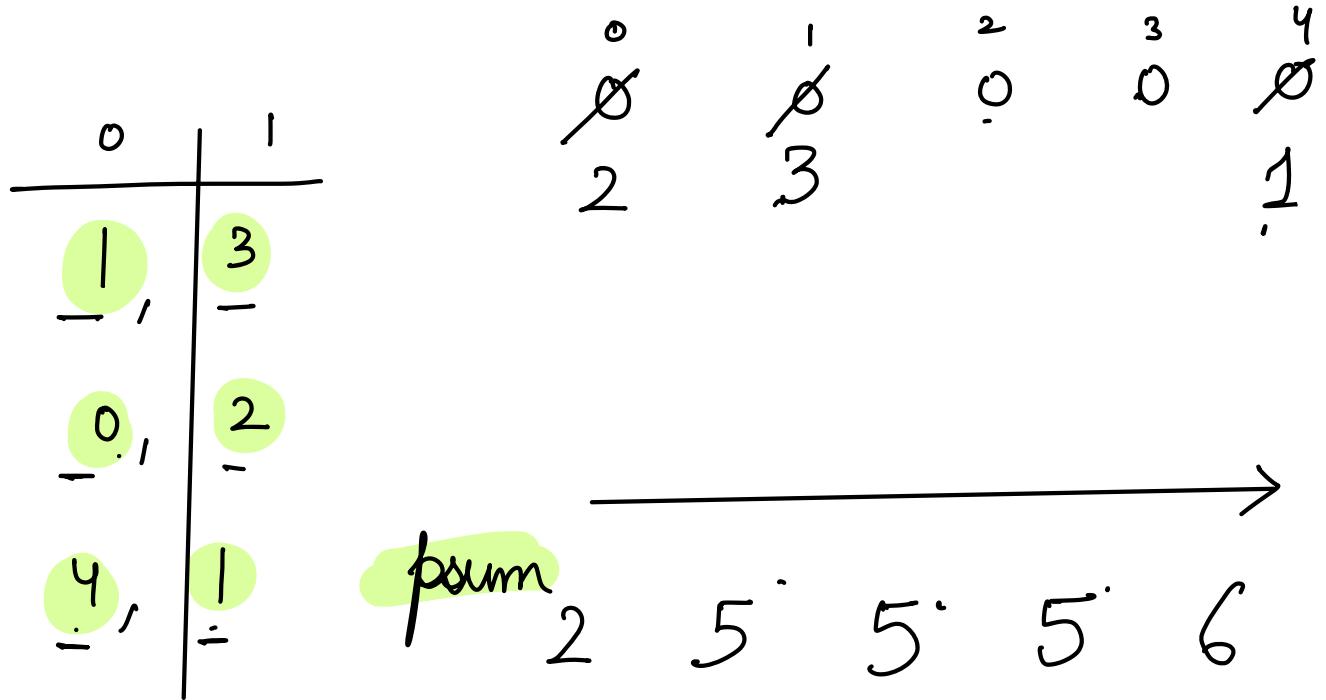


$\text{psum} = a_0 + a_0 + a_0 + a_0 + a_0$
 $\quad \quad \quad + a_1 + a_1 + a_1 + a_1 + a_1$
 $\quad \quad \quad a_1 + a_2 + a_2 + a_2 + a_2$
 $\quad \quad \quad + a_3 + a_3 + a_3 + a_3 + a_3$
 $\quad \quad \quad + a_4$



$\text{psum} = 0 \ 0 \ 5 \ 5 \ 5$

An arrow points to the right under the sequence of numbers.



Code : $\text{arr} = 0$

```
for ( i=0 ; i < Q.length ; i++ ) {
    ind = Q[i][0]
    val = Q[i][1]
    arr[ind] = arr[ind] + val
```

$\text{arr}[ind]$

```
psum[0] = arr[0]
for ( i=1 ; i < N ; i++ ) {
    psum[i] = psum[i-1] + arr[i]
```

}

psum

Doubts :

0, 2

1, 3

0, 5

0 0 0 0
+2 +2 +2 +2

+3 +3 +3

+5 +5 +5 +5

7 10 10 10

\emptyset
~~+2~~
7

\emptyset
3

0 0

7 10 10 10



Follow Up Question



< Question > : Initially all elements of an arr[N] are 0. Given Q queries.

Every query contains [s, e, val]. Increment elements from s to e by val.

Return the final state of arr[].

s — e

arr[10] → [0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0]
0 1 2 3 4 5 6 7 8 9

+3 / +3 / +3 / +3
-3 / -3 / -3 / -3 / -3

+4 +4 +4 +4 +4 +4 +4 +4 +4

Queries → 3

s	e	val	0	4	1	4	4	4	4	1	4	4	4
3	6	3											
2	7	-3											
1	9	4											
-	-	-											

1 ≤ N ≤ 10⁵

1 ≤ Q ≤ 10⁵

q (s, e, val)

s e
+ val



Quiz :

$\text{arr}[8] \rightarrow [0, 0, 0, 0, 0, 0, 0, 0]$

0 1 2 3 4 5 6 7

3 3 3 3

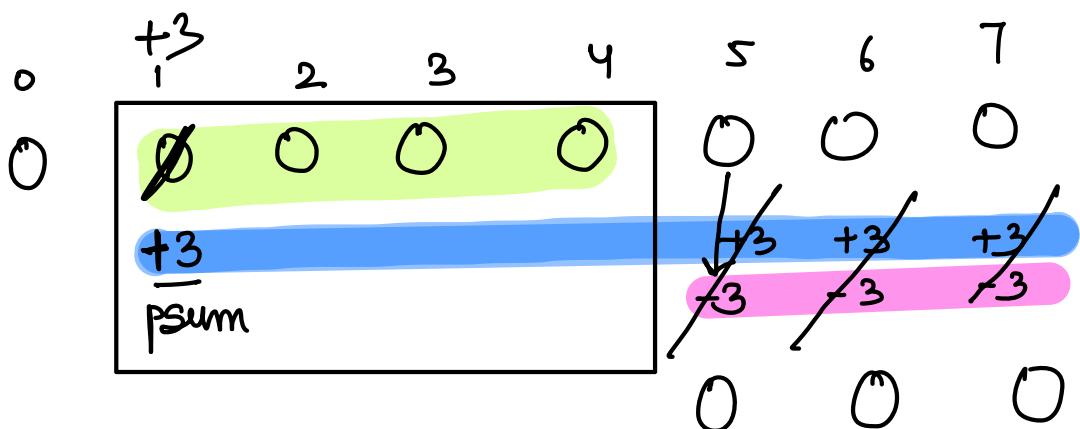
-1 + -1 -1 -1 -1.

Queries $\rightarrow 4$

+4
+3 +3 +3

si	ei	val
----	----	-----

1	4	3
0	5	-1
2	2	4
4	6	3



s }
e }

arr[s] += val

arr[e+1] -= val



if ($e+1 < N$)

Code

T

for every query \rightarrow

$S^{O(N-1)}, E^{O(N-1)}, val$

arr[s] += val

if ($e+1 < N$)

arr[e+1] -= val

Q

N [apply psum on arr

TC = $O(Q+N)$

J $O(N)$
SC $O(1)$

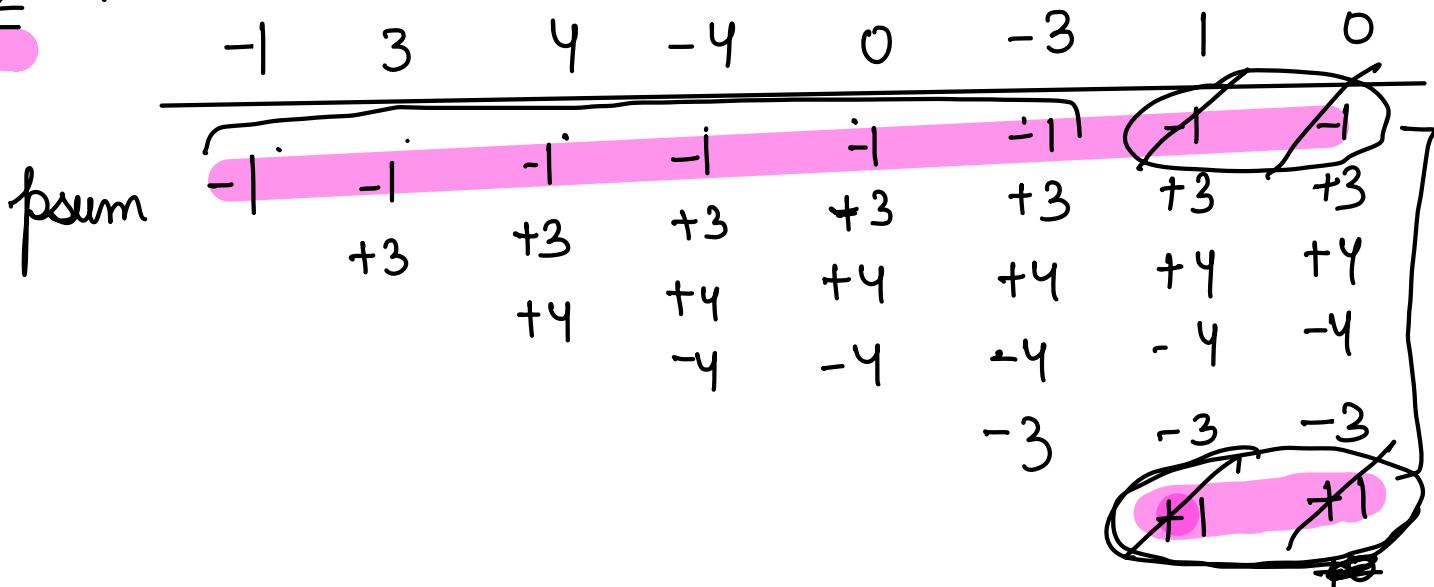


0	1	2	3	4	5	6	7
-1	+3				-3		
∅	∅	∅	∅	0	∅	∅	0
		+4	-4			1	

1 4 3
.

0 5 -1
.

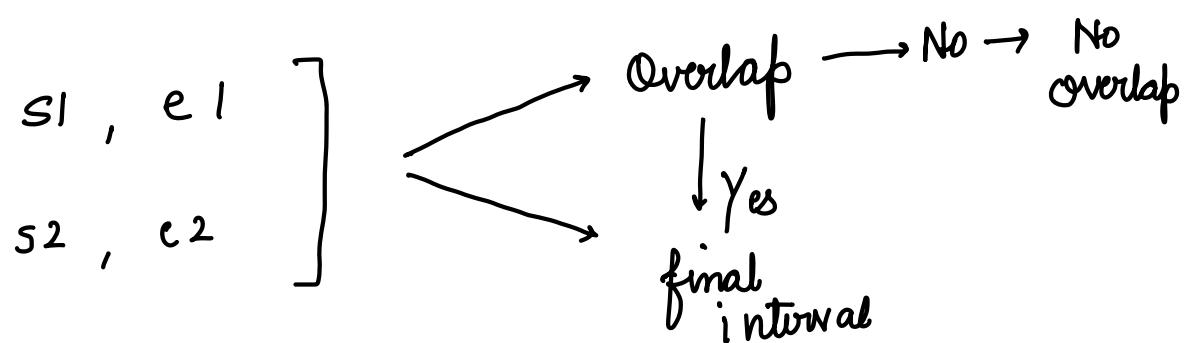
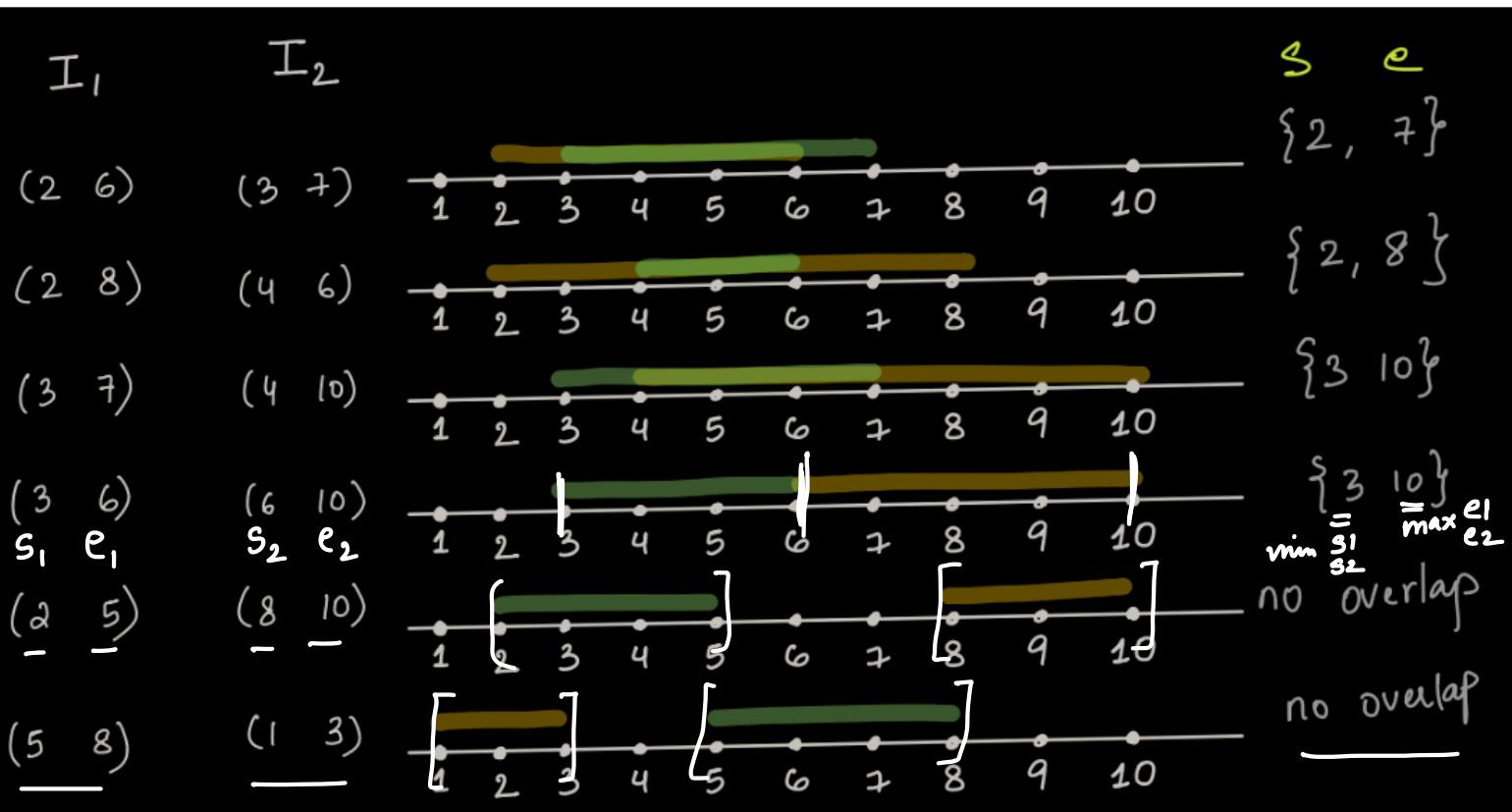
2 2 4



$$\text{arr}[s] = \text{arr}[s] + \text{val}$$

merge intervals

$$I_1 = (2, 6) \quad I_2 = (3, 7) \quad = \quad [2, 7]$$



s_1 , e_1

s_2 , e_2

s_3 , e_3

s_4 , e_4

s_2 , e_2

s_3 , e_3

s_4 , e_4

s_5 , e_5

No overlap condition



$$e_1 < s_2$$

or



$$e_2 < s_1$$

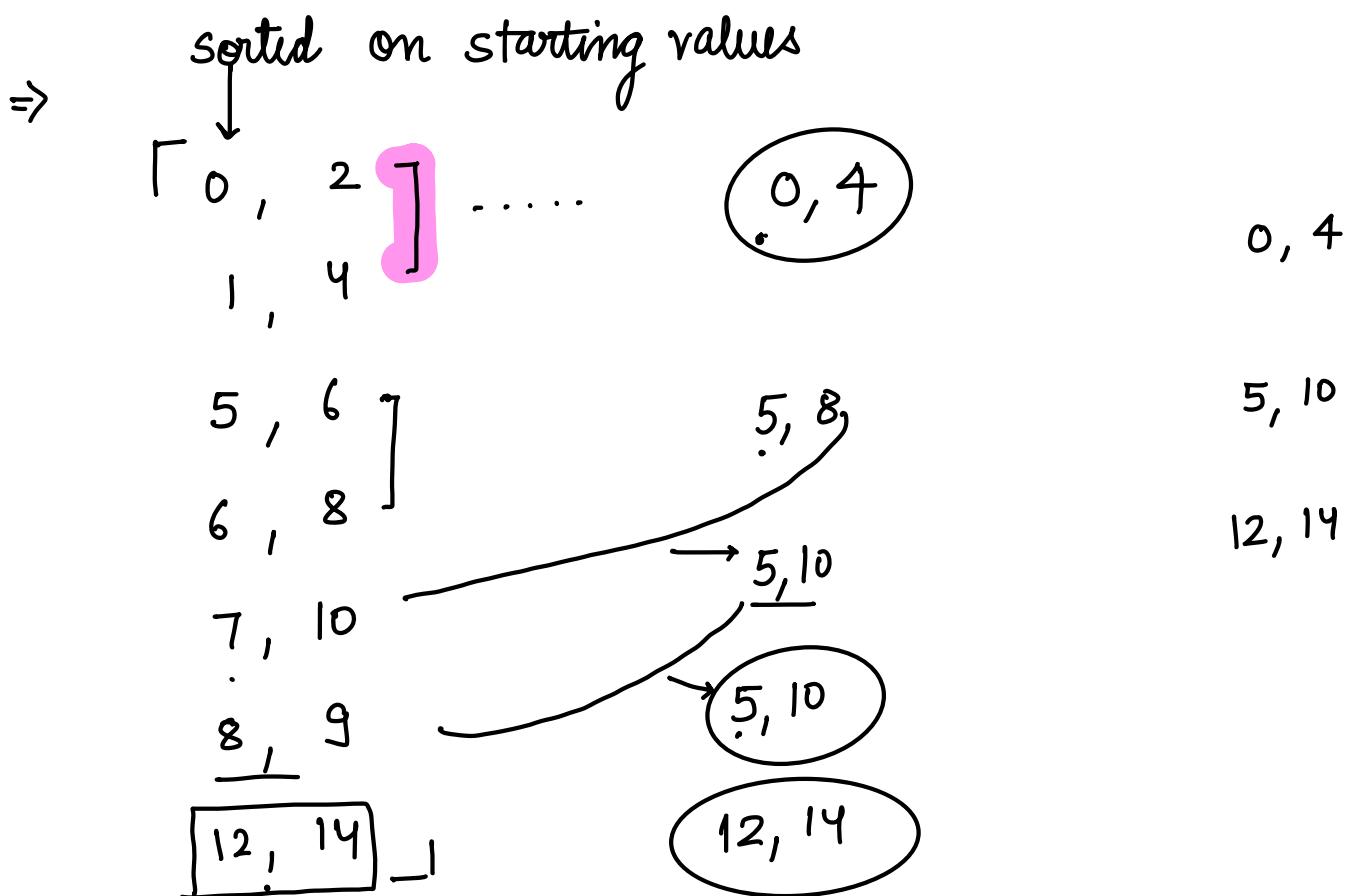
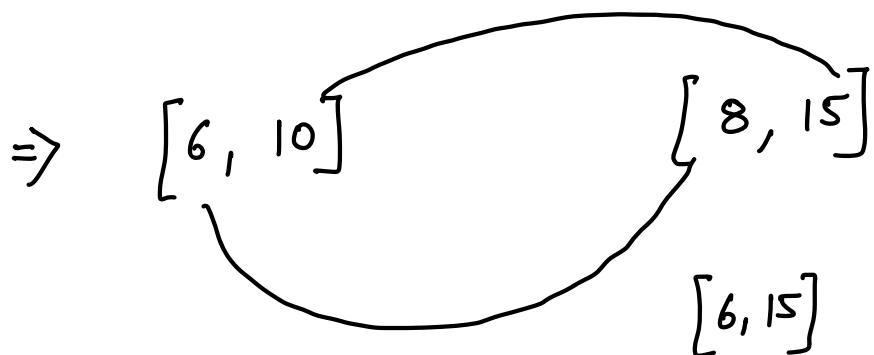
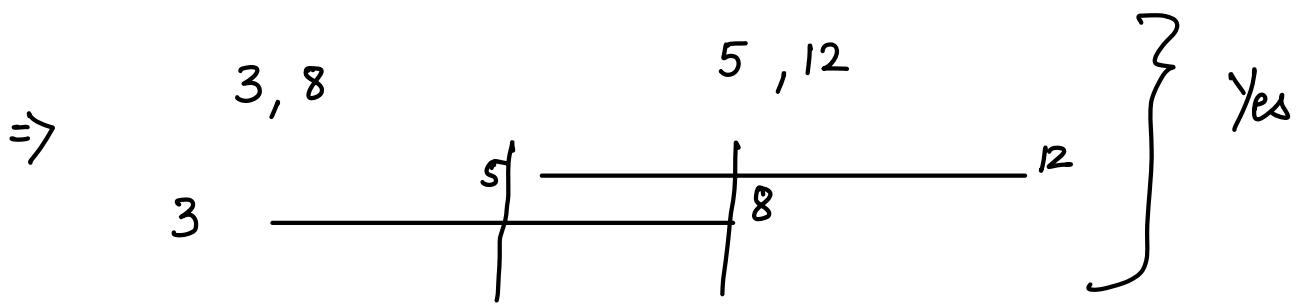
if ($e_1 < s_2$ || $e_2 < s_1$)
No overlap

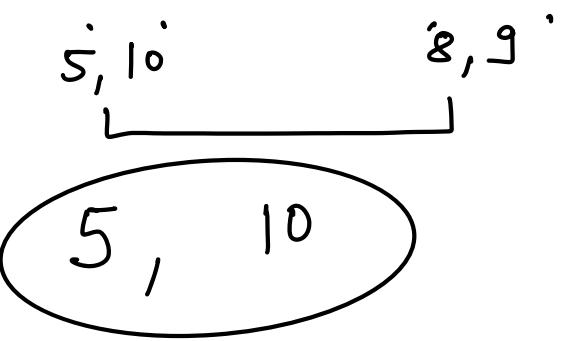
else

$$s = \min(s_1, s_2)$$

$$e = \max(e_1, e_2)$$

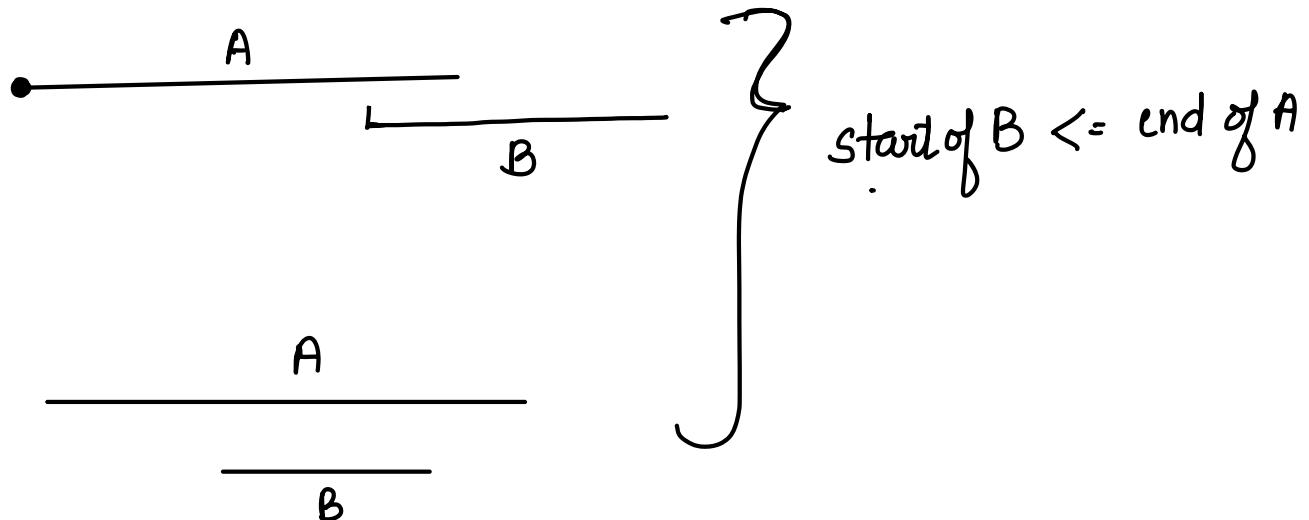




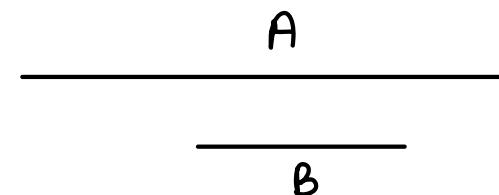


2D array is sorted on starting values

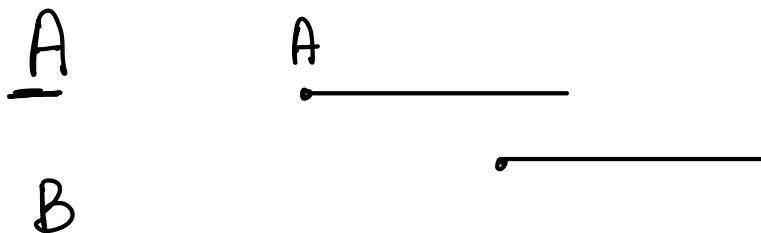
a)

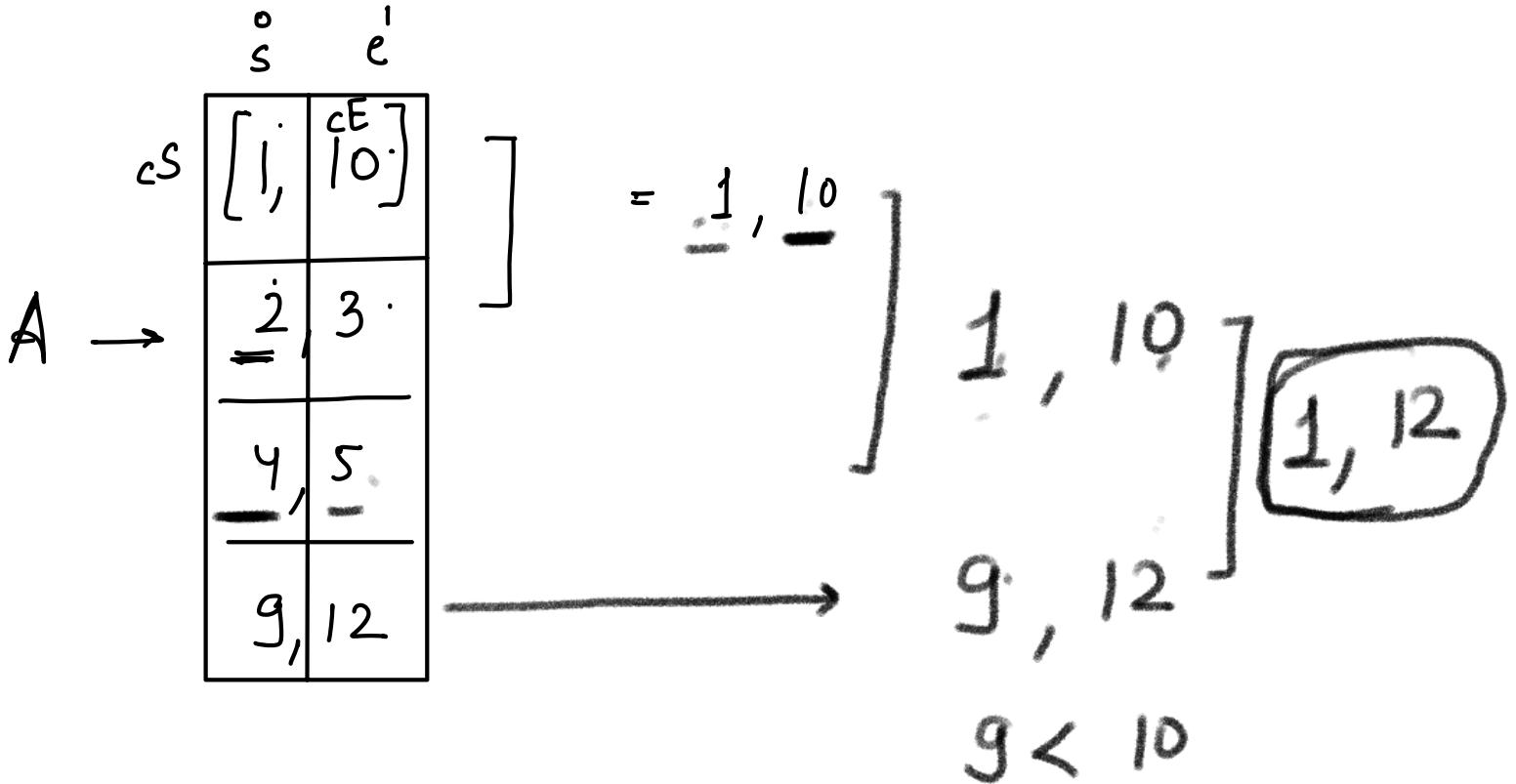


b)



c)





Outline \rightarrow

2D array is sorted on starting value

$$cS = A[0][0] \quad cE = A[0][1]$$

for ($i = 1 \dots i < N \dots i++$) {

if ($\underline{A[i][0]} <= cE$)

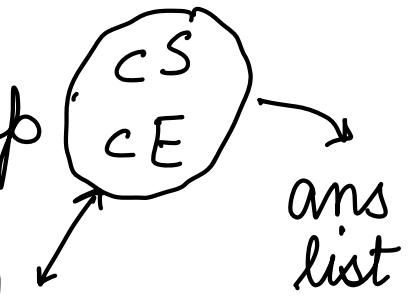
overlap \rightarrow cS
 cE

else

No overlap

$$CS = A[i][0]$$

$$CE = A[i][1]$$



}

return ans

Doubts

$$O(Q + N)$$

2 variables

$$Q = 2, N = 10000$$

$$Q = 10000, N = 5$$

$$Q = 1000, N = 5000$$

$$1 - N = 10^5$$

$$1 - Q = 10^5$$

worst case

$$\begin{array}{r} 10^5 \\ + 10^5 \\ \hline 2 * 10^5 \end{array}$$

$$O(N) \rightarrow O(N)$$

$$N = 10^5$$

$$\begin{array}{r} 2 * 10^5 \\ = \\ 10^5 \end{array}$$

$$O(N^2) \leftrightarrow O(N^2)$$

$$O\left(\frac{5000}{N} N\right) = O(N)$$

5000 N
 $\frac{5 \times 10^8}{N}$ >> $\frac{N}{10^5}$

✓ coefficients are small } →
 2
 3
 4

~~$\frac{N}{2 \times 10^5}$~~

$\frac{N}{10^5}$