

Q

Given an array of size  $N$ .

Google

Find the max sum of any contiguous chunk of elements. (Max sum of any subarray)

Facebook

Dream - ii  
Linked - ii

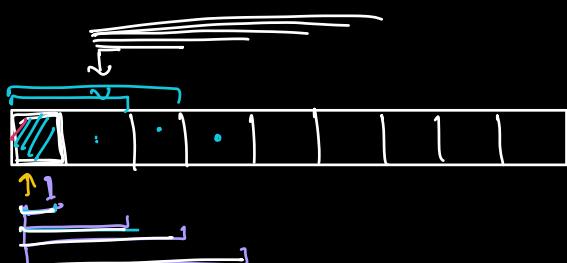
$$-2, \underbrace{3, 4, -1, 5,}_{\Sigma = 11} -10, 7$$

$$-3, \underbrace{4, 6, 8,}_{\Sigma = 18} -10, 2, 7$$

### Brute Force

$$TC = O(N^3)$$
$$SC = O(1)$$

```
manSum = INT-MIN / A[0]
for (i=0; i<N; i++) {
    for (j=i; j<N; j++) {
        sum = 0;
        for (k=i; k<=j; k++) {
            sum += A[k];
        }
        if (sum > manSum) {
            manSum = sum;
        }
    }
}
```



## Carry Forward

```

maxSum = -∞ / A[0];
for(i=0; i<N; i++) {
    sum = 0;
    for(j=i; j<N; j++) {
        sum = sum + A[j];
        maxSum = max(sum, maxSum);
    }
}

```

TC : O(N<sup>2</sup>)

SC : O(1)

## Test Case I

2, 4, 8, 10, 3		-2, -4, -8, -10, -3
----------------	--	---------------------

## Test Case II

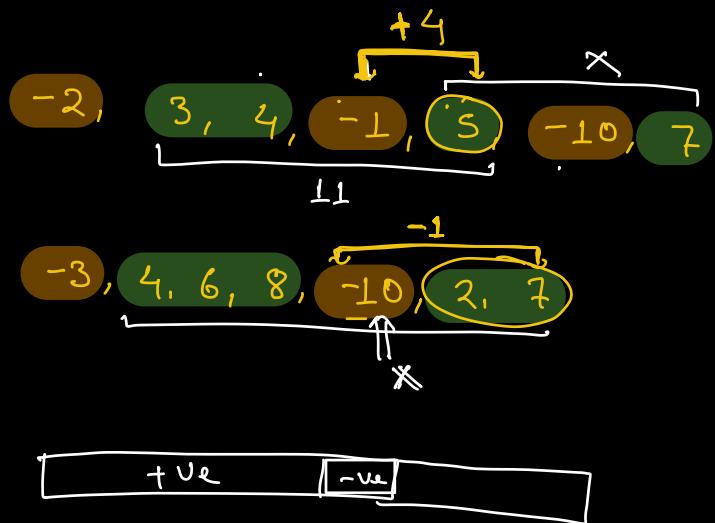
-10, -1, 2, 4, 8, 10, 3, -7, -9
---------------------------------

The answer will never have -ve elements in the corner. (assuming that array has +ve no.)



## Test Case III





$\boxed{3, \underline{4}, -1, 5,}$   
 $\boxed{3, 4, \underline{-8}, 5}$   
 $\boxed{3, 4, \underline{-6}, 5}$

Cum-Sum	5, 6, 7, -3, 2, -10, -12, 8, 12, 21, -4, 7
	5 11 18 15 17 7 $\cancel{-5} \Rightarrow 0$ 8 20 41 37 44
max-Sum	5 11 18 18 18 18 18 18 20 41 41 <u>44</u>

Sum	-20, 10, -20, -12, 6, 5, -3, 8, -2
	$\cancel{-20} \Rightarrow 0$ 10 $\cancel{-10} \Rightarrow 0$ $\cancel{-12} \Rightarrow 0$ 6 11 8 16 14
max-Sum	-20 10 10 10 11 11 16 16
	<u>16</u>

Kadane's  
Algo

manSum = INT\_MIN or A[0];    sum = 0;

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

sum += A[i];

manSum = max(sum, manSum);

    if (sum < 0)

sum = 0;

}

return manSum;

TC : O(N)

SC : O(1)

HW : Find the s & e index  
of the max sum sub-array.

VMware  
Goldmensch

Direct i

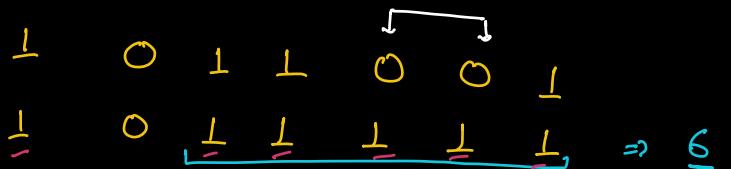
Hard

Q Given a binary array (all elements are 0/1)

You are allowed to perform AT MOST 1 flip  
Operation on any subarray.

$$\begin{matrix} 0 & \rightarrow & 1 \\ 1 & \rightarrow & 0 \end{matrix}$$

Return the max no of 1's that can be  
achieved after performing this op.



$$\begin{matrix} \perp & \perp & \perp & \overbrace{\perp \ 0}^1 & \perp & 0 \\ \perp & \perp & \perp & \perp & \perp & 0 \\ \perp & \perp & \perp & \perp & 0 & \perp \end{matrix} \Rightarrow 5$$

$$0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$$

flip (5-7)

$$0 \ 0 \ 0 \ 1 \ 1 \ \perp \ \perp \ \perp \ \perp \ \Rightarrow 6$$

flip (0-8)

$$\perp \ \perp \ \perp \ 0 \ 0 \ \perp \ \perp \ \perp \ 0 \ \Rightarrow 6$$

flip (0-7)

$$\perp \ \perp \ \perp \ 0 \ 0 \ \perp \ \perp \ \perp \ 1 \ \Rightarrow 7$$

Test Cases I

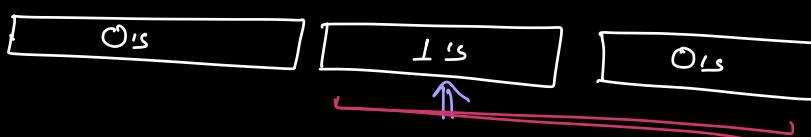
0 0 0 0 0 0		1 1 1 1 1
-------------	--	-----------

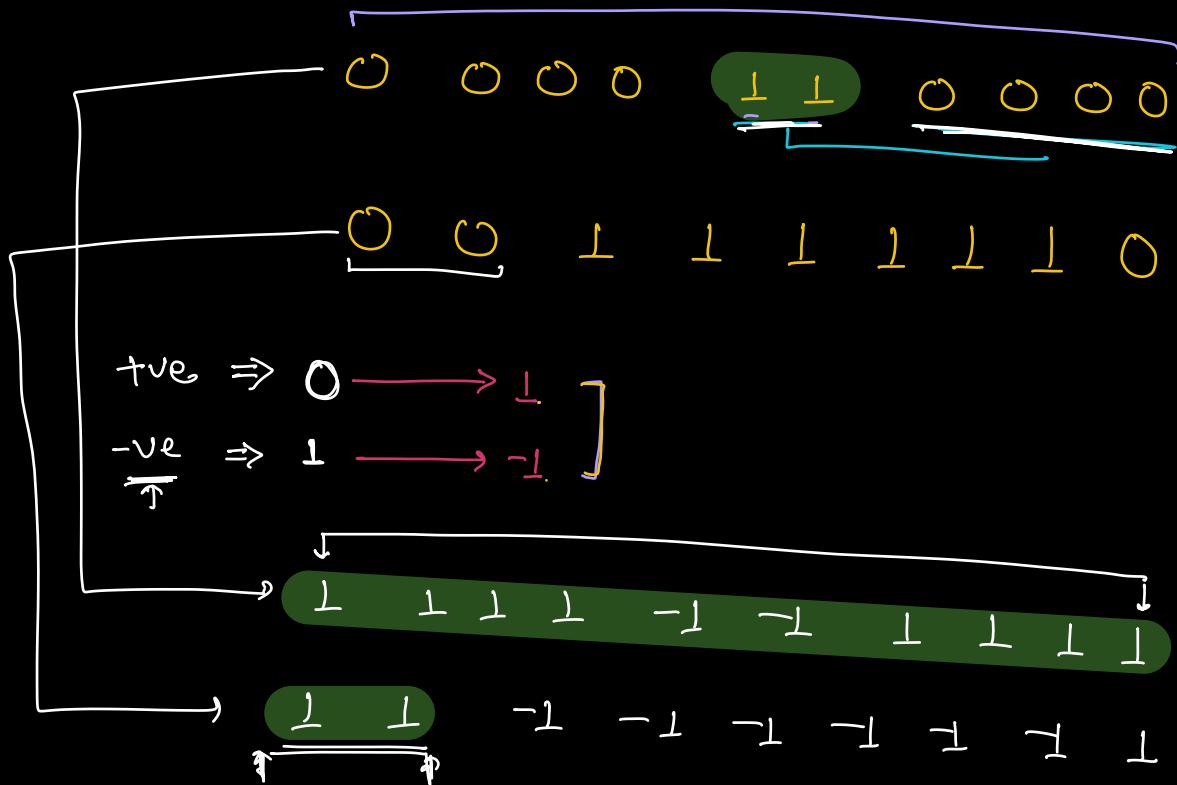
Test Case II

1 1 1 0 0 0 0 0 1 1 1 1
-------------------------

{ The subarray will never have 1's in the corner. (assuming that array has 0's) }

Test Case III





Step I : a) Convert all 1 to -1  
 b) Convert all 0 to 1

Step II : Apply Kadane's Algo & get the  
 subarray with max sum.  
 $\hookrightarrow (s, e)$

Step III : Count the no of 1's after flipping  
 subarray from s to e.

Break till 10:40 pm

Q Given an array of size N with all elements as 0.  
 $\forall i \quad A[i] = 0;$

Given Q queries of index (i) & value (v)

Add the val to all the array elements starting from  
 $i.$  Return final state of array

A :	0	0	0	0	0	0	0	i	v
	Index	1	2	3	4	5	6	val	
0	0	3	3	3	3	3	3	1	3
0	0	3	3	3	5	5	5	4	2
0	0	3	4	4	6	6	6	2	1

$$A : 0 \ 3 \ 4 \ 4 \ 6 \ 6 \ 6$$

### Brute Force

Iterate over all query.

TC:  $O(QN)$

SC:  $O(1)$

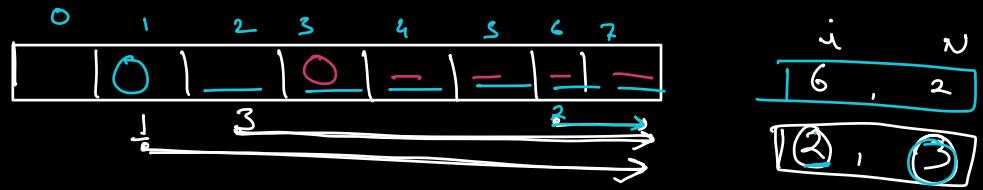
$\text{for } (j=0; j < Q; j++) \{ \Rightarrow Q$

// read jth query (index, val)

$\text{for } (i = \text{index}, i < N; i++) \{ \Rightarrow N$   
 $A[i] += \text{val},$

}

]



$\begin{matrix} 1 \\ 6 \end{matrix}, \quad \begin{matrix} 2 \\ 3 \end{matrix}, \quad \begin{matrix} 3 \\ 1 \end{matrix}$

A

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

PS	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$
	+	+	+	+	+
	$a_1$	$a_1$	$a_1$	$a_1$	$a_1$
	$q_1$	$q_1$	$q_1$	$q_1$	$q_1$
	$q_2$	$q_2$	$q_2$	$q_2$	$q_2$
		$q_2$	$q_2$	$q_2$	$q_2$
			$q_3$	$q_3$	$q_3$
				$q_3$	$q_3$
					$q_4$

A

0	1	2	3	4	5	6	7
0	2	4	2	0	1	0	0

0 2 6 8 8 9 9 9

$i, v$	0	1	2	3	4	5	6	7
2, 3	0	0	3	3	3	3	3	3
5, 1	0	0	3	3	3	4	4	4
3, 2	0	0	3	5	5	6	6	6
2, 1	0	0	4	6	6	7	7	7
1, 2	0	2	6	8	8	9	9	9

Step I : for every query.  
 $A[i] += val;$  ]  $\Rightarrow O(Q)$

Step II : Take the prefix sum ]  $\Rightarrow O(N)$

TC :  $O(N+Q)$

What if ?

30 min

Google

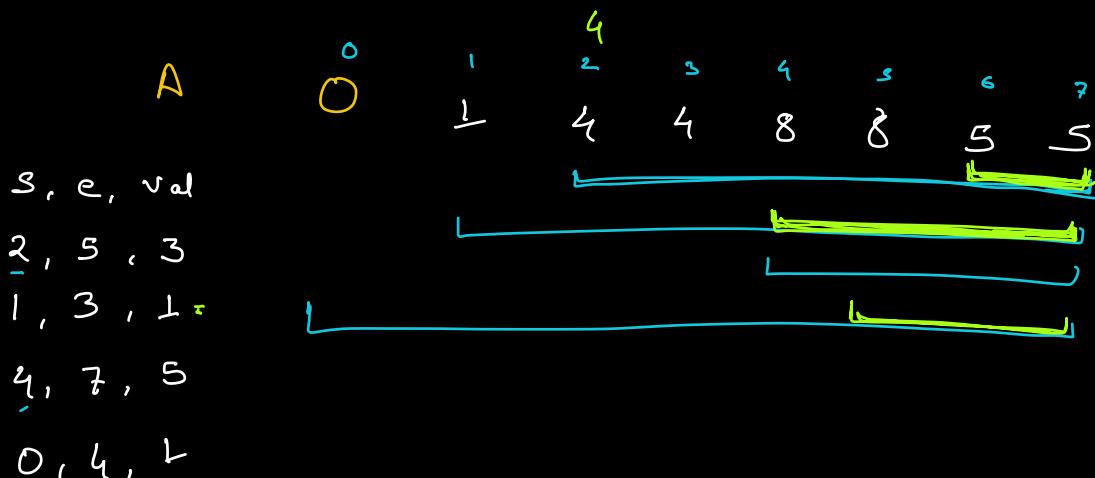
Q

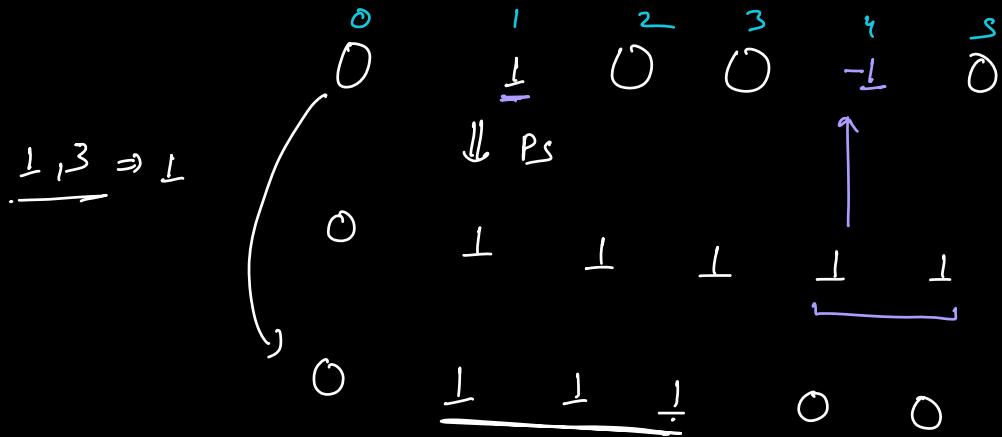
Given an array which initially has all 0's.

Given Q queries;

s, e, val

Add val to all elements in the subarray from s to e.





$A : \begin{matrix} 0 & 1 & 2 & 3 & 1 & \perp & 0 & -3 & 3 \end{matrix}$

s, e, val

$3, 6 \perp \longrightarrow A[3] += \perp ; A[7] += (\sim 1);$

$\perp, 4 \ 2 \longrightarrow A[1] += 2 ; A[5] += (\sim 2)$

$2, 5 \ 3 \longrightarrow A[2] += 3 ; A[6] += (\sim 3)$

$4, 7 \perp \longrightarrow A[4] += \perp ;$

$5 \ 6, 2 \longrightarrow A[5] += 2 ; A[7] += (\sim 2)$

Take Prefix Sum  $\implies$

0    2    5    6    7    7    4    7

Facebook

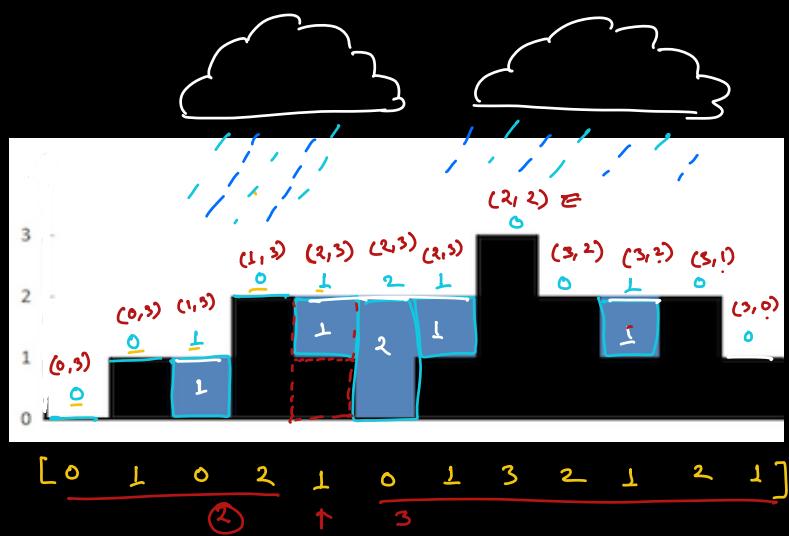
Google

Amazon

## Q Rain water trapping

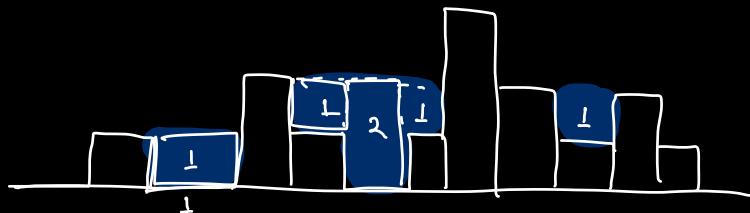
Given an array of non-negative no. representing the heights of buildings.

Return the total amount of water that will be trapped b/w the buildings if it is rainy Continuously.



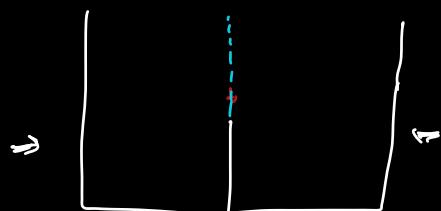
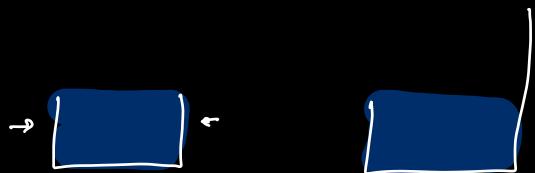
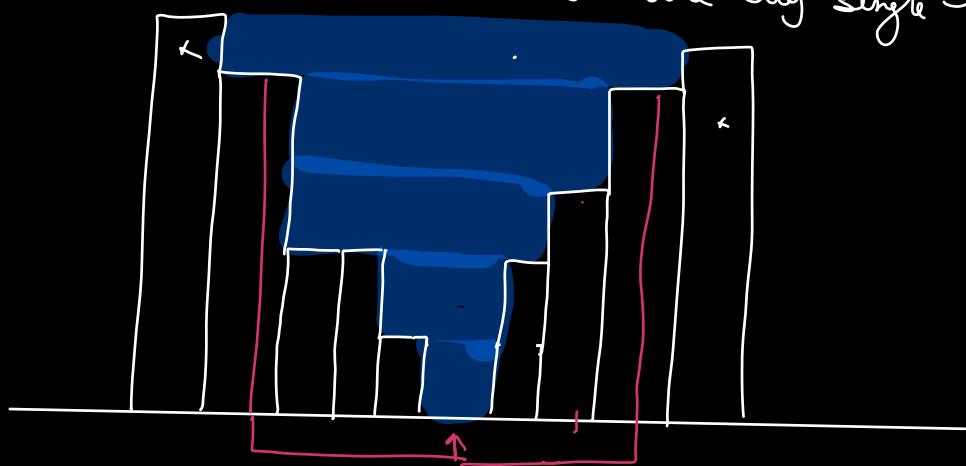
6 units of water.

$[0 \quad 1 \quad 0 \quad 2 \quad 1 \quad 0 \quad 1 \quad 3 \quad 2 \quad 1 \quad 2 \quad 1]$



$$1 + 1 + 2 + 1 + 1 = \underline{6 \text{ uni}}$$

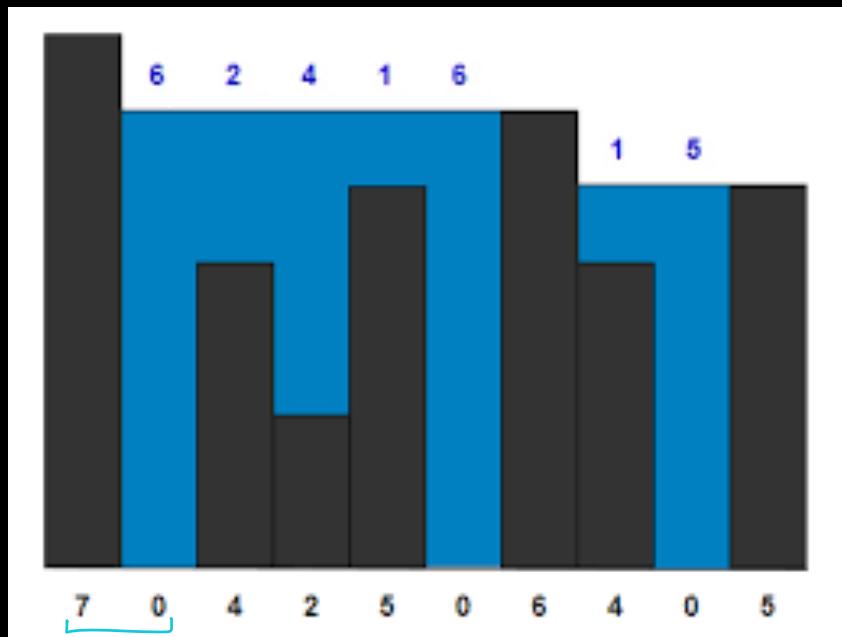
Total amount of water trapped = Sum of the heights of water columns above every single boundary.



Height of water column above a boundary at index  $i$

$$Ht = \min(\underline{\text{leftMan}}, \underline{\text{rightMan}}) - A[i]$$

$$Ht < 0 \Rightarrow Ht = 0$$



	7	0	4	2	5	0	6	4	0	5
Left Max	0	7	7	7	7	7	7	7	7	7
Right Max	6	6	6	6	6	6	5	5	5	0

$$ht = \min(LM, RM) \quad 0 \quad 6 \quad 2 \quad 4 \quad 1 \quad 6 \quad 0 \quad 1 \quad 5 \quad 0$$

-  $A(i)$

0 - 5  
 $\leftarrow s\right)$

TC :  $O(N)$

$\overbrace{\quad \quad \quad}^N + \overbrace{\quad \quad \quad}^N + \overbrace{\quad \quad \quad}^N$   
 Left Max      Ans Calculation  
 Right M

SC :  $O(N)$

$\overbrace{\quad \quad \quad}^{O(1)} \text{ SC}$   
Hw