Q → Given an integer array s.t ∀ i A[i] = 0.
Return the final A[] after performing multiple queries.

Query → (i, x) → Add x to all numbers from A[i] to A[N-1].
i ⟶ end

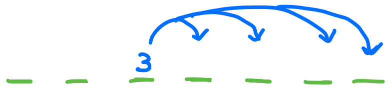
0 1 2 3 4 5 6
A = [0 0 0 0 0 0 0]
+3 +3 +3 +3 +3 +3
+2 +2 +2
+1 +1 +1 +1
[0 3 3 4 6 6 6] ✓

Queries
(1, 3)
(4, 2)
(3, 1)

Bruteforce → ∀query add x from i to (N-1).
O(Q) * O(N) ⟶ TC = O(Q * N)
SC = O(1) ✓

P[i] = P[i-1] + A[i] ✓
A[i] += A[i-1]

i ⟶ N-1 ⟵ prefix sum left to Right
i ⟵ N-1 suffix sum Right to Left



Queries
(1, 3)
(4, 2)
(3, 1)

0 1 2 3 4 5 6
A = [0 ~~0~~ 0 ~~0~~ ~~0~~ 0 0]
3 1 2
[0 3 3 4 6 6 6] ✓

Queries one by one
TC = O(Q * N)
(travel for every query)

← combine all queries.

I/p → I = [1 4 3]
X = [3 2 1]

0 1 2 3 4 5 6 7
A = [0 0 ~~0~~ ~~0~~ 0 ~~0~~ 0 0]
~~3~~ ~~8~~ -1
-2 10
[0 0 -2 8 8 7 7 7] ✓

Queries
(2, 3)
(5, -1)
(3, 8)
(2, -5)
(3, 2)

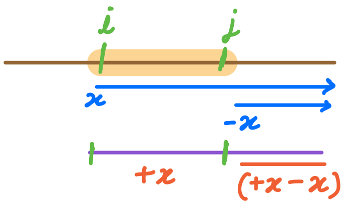
TC = O(Q + N)
SC = O(1) ✓

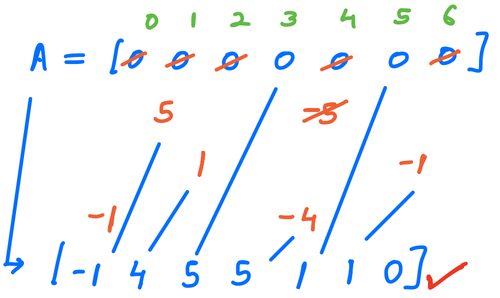
Q → Given an integer array s t ∀ i A[i] = 0.
Return the final A[ ] after performing multiple queries.

Query → (i, j, x) → Add x to all numbers from A[i] to A[j].

i → j $\boxed{i <= j}$

$$\begin{array}{cccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ A = [& 0 & 0 & 9 \;{\scriptstyle\hookrightarrow -9} & 0 & 0 & 0 & 0\;] \\ & & +5 & +5 & +5 & & & \\ & & & +1 & +1 & +1 & +1 & \\ & -1 & -1 & -1 & -1 & & & \\ \hookrightarrow [& -1 & 4 & 5 & 5 & 1 & 1 & 0\;] \checkmark \end{array}$$





Query

(1, 3, 5) → (1, 5) (3+1, −5)
(2, 5, 1) → (2, 1) (5+1, −1)
(0, 3, −1) → (0, −1) (3+1, −(−1) = 1)

$$\begin{array}{cccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ A - [& 0 & 0 & 0 & 0 & 0 & 0 & 0\;] \end{array}$$

(1, 3, 5) +5 +5 +5 +5 +5 +5 (1, 5)
−5 −5 −5 (3+1, −5)

Query → Q=3

(1, 3, 5)
(2, 5, 1)
(0, 3, −1)

$$\begin{array}{cccc} & 0 & 1 & 2 \\ I = [& 1 & 2 & 0\;] \\ J = [& 3 & 5 & 3\;] \\ X = [& 5 & 1 & -1\;] \end{array}$$

$$\begin{array}{cccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ A - [& 0 & 0 & 0 & 0 & 0 & 0 & 0\;] \\ & & & & +5 & +5 & +5 & +5 \end{array}$$

Query
(3, 6, 5)
↙
(3, 5) (6+1, −5)
7

```
for k → 0 to (Q-1)
    i = I[k]
    j = J[k]
    x = X[k]
    A[i] += x
    A[j+1] -= x        → Index out of bound
for i → 1 to (N-1)     } prefix sum
    A[i] += A[i-1]
```

```
return A
```

```
for k → 0 to (Q-1)
    i = I[k]
    j = J[k]
    x = X[k]
    A[i] += x
    if (j+1 < N) A[j+1] -= x
for i → 1 to (N-1)          } prefix sum
    A[i] += A[i-1]
return A
```

$TC = O(Q+N)$

$SC = \underline{O(1)}$

Q → Given an integer array, find max value of $f(i,j)$.

$$f(i,j) = (A[i] - A[j]) \quad \forall\, i,j \text{ index}$$

$A = [\overset{0}{1} \;\; \overset{1}{3} \;\; \overset{2}{-2}]$

$i \to 0$ to $N-1$

$j \to 0$ to $N-1$

i	j	A[i] − A[j]
0	0	1 − 1 = 0
0	1	1 − 3 = −2
0	2	1 − (−2) = 3
1	0	3 − 1 = 2
1	1	3 − 3 = 0
1	2	3 − (−2) = $\underline{5}$ (Ans)
2	0	−2 − 1 = −3
2	1	−2 − 3 = −5
2	2	−2 − (−2) = 0

Bruteforce → $TC = O(N^2)$

$SC = \underline{O(1)}$

$$f(i,j) = A[i] - A[j]$$

high A[i] ⇒ increase $f(i,j)$

low A[j] ⇒ increase $f(i,j)$

$f = x - y$

↑ ↓

$\max(A) - \min(A) = 3 - (-2) = \underline{5}$ ✓

$$\boxed{\text{Ans} = \max(A) - \min(A)}$$

→ sort → $TC = O(N \log(N))$

→ travel to find max & min → $TC = \underline{O(N)}$

$A = [1 \quad 3 \quad 5 \quad 2] \longrightarrow 5 - 1 = \underline{4}$ $\qquad$ $SC = \underline{O(1)}$

$A = [-1 \quad -3 \quad -5 \quad -2] \longrightarrow -1 - (-5) = \underline{4}$

Q → Given an integer array, find max value of $f(i,j)$.

$$f(i,j) = |A[i] - A[j]| + |i - j| \qquad \forall\, i,j \text{ index}$$

$$A = [\overset{0}{1} \quad \overset{1}{3} \quad \overset{2}{-2}]$$

$|x| \to x$ if $x >= 0$
$\quad\;\; \to -x$ if $x < 0$

$|-2| = -(-2) - 2$

$|5| = 5$

i	j	$\lvert A[i] - A[j] \rvert$	$\lvert i-j \rvert$	$f(i,j)$
0	0	$\lvert 1-1 \rvert = 0$	$\lvert 0-0 \rvert = 0$	$0+0=0$
0	1	$\lvert 1-3 \rvert = 2$	$\lvert 0-1) = 1$	$2+1=3$
0	2	$\lvert 1-(-2) \rvert = 3$	$\lvert 0-2 \rvert = 2$	$3+2=5$
1	0	$\lvert 3-1 \rvert = 2$	$\lvert 1-0) = 1$	$2+1=3$
1	1	$\lvert 3-3 \rvert = 0$	$\lvert 1-1) = 0$	$0+0=0$
1	2	$\lvert 3-(-2)) = 5$	$\lvert 1-2 \rvert = 1$	$5+1=\underline{6}$ (Ans)
2	0	$\lvert -2-1) = 3$	$\lvert 2-0 \rvert = 2$	$3+2=5$
2	1	$\lvert -2-3 \rvert = 5$	$\lvert 2-1) = 1$	$5+1=\underline{6}$
2	2	$\lvert -2-(-2) \rvert = 0$	$\lvert 2-2 \rvert = 0$	$0+0=0$

Bruteforce ⇒ $TC = O(N^2)$
$SC = \underline{O(1)}$

Observations → 1) $(i == j) \Rightarrow f(i,j) = \underline{0}$ ✓ → ignore ✓

2) $\min(f(i,j)) = \underline{0}$ ✓

$$f(i,j) = \underbrace{|A[i] - A[j]|}_{>=0} + \underbrace{|i - j|}_{>=0} \qquad \forall\, i,j \text{ index}$$

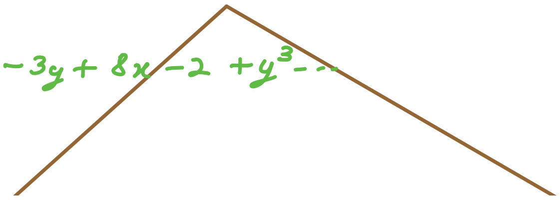
Only check cases where $(i > j)$ or $(j > i)$. ✓

3) $f(i,j) = f(j,i)$

Ans $= f(2,5) \longrightarrow f(5,2)$

$$f(i,j) = |A[i] - A[j]| + (i - j) \qquad \forall\; \boxed{i > j} \rightarrow \boxed{i - j > 0}$$

$f(x,y) = x^2 - 3y + 8x - 2 + y^3 - -$



$|x| \to x$ if $x >= 0$
$\quad\;\; \to -x$ if $x < 0$

$(A[i] >= A[j])$

$f(i,j) = A[i] - A[j] + i - j$

$\downarrow$

$f(i,j) = \boxed{(A[i]+i)} - \boxed{(A[j]+j)}$ ✓

max min

$(A[i] < A[j])$

$f(i,j) = -A[i] + A[j] + i - j$

$f(i,j) = \boxed{(A[j]-j)} - \boxed{(A[i]-i)}$ ✓

max min

$f = x - y$

$\uparrow \quad \downarrow$

$$A = \begin{array}{ccc} 0 & 1 & 2 \\ [1 & 3 & -2] \end{array}$$

$(A[i]+i) \rightarrow \quad 1 \quad 4 \quad 0$

max min

$4 - 0 = \underline{4}$

$$A = \begin{array}{ccc} 0 & 1 & 2 \\ [1 & 3 & -2] \end{array}$$

$(A[i]-i) \quad 1 \quad 2 \quad -4$

max min

$2 - (-4) = \underline{6}$

$\text{Ans} = \max(4, 6) = \underline{6} \leftarrow$

$$A = \begin{array}{ccc} 0 & 1 & 2 \\ [1 & 1 & 1] \end{array}$$

$f(i,j) = \boxed{(A[i]+i)} - \boxed{(A[j]+j)}$

$A[i]+i \rightarrow$ ① 2 ③ $\quad 3 - 1 = \underline{2}$

$f(i,j) = \boxed{(A[j]-j)} - \boxed{(A[i]-i)}$

$A[i]-i \rightarrow$ ① 0 ⊖1 $\quad 1-(-1) = \underline{2}$

$\text{Ans} = \underline{2}$ ✓

$$A = \begin{array}{ccc} 0 & 1 & 2 \\ [1 & 1 & 1] \end{array}$$

$f(i,j) = \underbrace{|A[i] - A[j]|}_{\underline{0}} + \underbrace{|i-j|}_{2 - 0 = \underline{2}} \rightarrow \text{Max} = \underline{2}$

max $A[i]+i$

min $A[i]+i$

max $A[i]-i$

min $A[i]-i$

$\Big\} \rightarrow$ $TC = O(N)$

$SC = \underline{\underline{O(1)}}$

$$A = \begin{array}{ccc} 0 & 1 & 2 \\ [① & 3 & ⊖2] \\ \cancel{i}\ i & & j \end{array}$$

11:05 PM

$f(i,j) = |A[i] - A[j]| + |i - j|$

$|1 - (-2)| + |0 - 2| = 3 + 2 = \underline{5}$ ✓

$TC = O(N)$

$$A = \begin{array}{ccc} 0 & 1 & 2 \\ [① & -4 & ⊖2] \\ \cancel{i} \rightarrow i & & j \end{array}$$

$f(0,2) \rightarrow \underline{5}$ ✓

$f(1,2) \rightarrow |-4-(-2)| + |1-2| = 2+1 = \underline{3}$

$f(0,1)$

$|1-(-4)| + |0-1| = 5+1 = \underline{6}$ ✓

$$A = \begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & 8 & 10 & 0 & 6 & 9 \end{array}$$

✓ ✓

$1 + 10 = \underline{11}$

$6 + 8 = \underline{14}$ ✓

Q → Find the max subarray sum ∀ subarrays in the array.

subarrays → continuous part of array

$$A = \begin{array}{ccc} 0 & 1 & 2 \\ 1 & 3 & -2 \end{array}$$

\# subarrays $= \frac{N*(N+1)}{2}$ ✓

[1] → 1

[1 3] → $\underline{4}$ (Ans)

[1 3 -2] → 2

[3] → 3

[3 -2] → 1

[-2] → -2

Bruteforce → ∀ subarrays calculate sum & store maximum.

$TC = O(N^2 * N) = O(N^3)$

↓ carry forward

$TC = \underline{O(N^2)}$ ✓ $SC = \underline{O(1)}$

```
ans = ~~0~~ → INT_MIN or A[0] ✓
for i → 0 to (N-1)          //start
    sum = 0
    for j → i to (N-1)      //end
        sum += A[j]
        ans = max(ans, sum)
return ans
```

$A = [-2\ \ -8\ \ -1]$ ✓

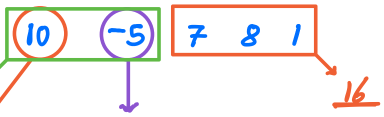
Ans = $\underline{-1}$

$-8 < -2 < \underline{-1}$ ✓

Observation → 1) $\forall i\ A[i] >= 0 \rightarrow Ans = \sum_{i=0}^{N-1} A[i]$ (include all elements)

2) $\forall i\ A[i] < 0 \rightarrow Ans = max(A)$ (exclud negative elements)

3)



not include -5 as it will decrease subarray sum.

cannot include 10 alone because → subarray is continuous.

→ include 10-5 = 5 ∵ it is increasing sum.

	L			R								
	0	1	2	3	4	5	6	7	8	9	10	11
A =	[10	-5	7	8	-11	-20 |	10	-3	-30 |	8	5	-6]
sum →	~~10~~	~~5~~	~~12~~	~~20~~	~~9~~	~~-11~~ ✓	~~10~~	~~7~~	~~-23~~ ✓	~~8~~	~~13~~	7

ans = ~~10~~ ~~12~~ 20 ✓

Kadane's Algo

```
ans = A[0]
sum = 0
for i → 0 to (N-1)
    sum += A[i]
    ans = max(ans, sum)
    if (sum < 0)
        sum = 0
return ans
```

TC = O(N) SC = O(1)

	✓ 0	✓ 1	✓ 2	✓ 3	
A =	[-2	-8	-1	-3]	→ Ans = -1

i	sum	ans
	0	-2 / INT_MIN
0	~~-2~~ 0	-2
1	~~-8~~ 0	-2
2	~~-1~~ 0	-1
3	~~-3~~ 0	-1 ✓

H.W → Find any one subarray with max sum i.e find L, R s.t. sum of subarray from L to R is max. ✓

	0	1	2	3	4
A =	[3	8	-20	8	3]
	✓			✓	