Q1) Given an array of size N, having all 0s and given q queries (i, val). For each query add val for all elements [i, N-1]

	0	1	2	3	4	5	6	7	8	9	10
	{0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0}
			+3	+3	+3	+3	+3	+3	+3	+3	+3
						+4	+4	+4	+4	+4	+4
								+6	+6	+6	+6
	0	0	3	3	3	7	7	13	13	13	13

i	val
2	3
5	4
7	6

A

```
for(i=0; i<q; i++){
    ind = Q[i][0]
    val = Q[i][1]
    for(j=ind; j<N; j++){
        A[j] += val
    }
}
```

→ Bottle neck

TC: O(NQ)
SC: O(1)

{a_0 a_1 a_2 a_3 a_q}

Pf sum array

{a_0	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

$a_0 \rightarrow [0, N-1]$
$a_1 \rightarrow [1, N-1]$
$a_2 \rightarrow [2, N-1]$

0	1	2	3	4	5	6	7	8	9	10
{0	0	0	0	0	0	0	0	0	0	0}
		+3								
					+4					
							+6			

i	val
2	3
5	4
7	6

0	1	2	3	4	5	6	7	8	9	10
{0	0	3	0	0	4	0	6	0	0	0}

Pf sum

{0 0 3 3 3 7 7 13 13 13 13}

```
for(i=0; i<q; i++){
    ind = Q[i][0]
    val = Q[i][1]
    A[ind] += val
}
```

TC: O(Q+N)
SC: O(1)

```
for(i=1; i<N; i++){
    A[i] = A[i-1] + A[i]
}
```

Google

Q2) Given an array of size N, having all 0s and given q queries (i, val). For each query add val for all elements [i, j]

0 1 2 3 4 5 6 7 8 9 10
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
+2 +2 +2 +2 +2
+1 +1 +1 +1 +1 +1
-3 -3 -3

[0 2 3 3 0 0 0 1 0 0 0]

i	j	val
1	5	2
2	7	1
4	6	-3

Brute force [still work]

0 1 2 3 4 5 6 7 8 9 10
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
+2 +2 +2 +2 +2 +2 +2 +2 +2
Need — dont need
-2 -2 -2 -2 -2 -2
0 0 0 0 0

i	j	val
1	5	2
2	7	1
4	6	-3

ind2 = N-1

```
for(i=0; i<q; i++){
    ind1 = Q[i][0]        val = Q[i][2]
    ind2 = Q[i][1]
    A[ind1] += val
    if (ind2 != N-1){
        A[ind2+1] -= val
    }
}

for(i=1; i<N; i++){
    A[i] = A[i-1] + A[i]
}
```

TC: O(N+Q)
SC: O(1)

0 1 2 3 4 5 6 7 8 9 10

{0 0 0 0, 0, 0, 0, 0, 0, 0, 0}

+2 -2

+1 -1

i	j	val
1	5	2
2	7	1
4	6	-3

-3 +3

0 2 1 0 -3 0 -2 3 -1 0 0

Ans: 0 2 3 0 0 0 -2 1 0 0 0

$\boxed{l \le r}$

Q3) Given an array, find max subarray sum.

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

Ans : 8

1) Find all subarray sums [Brute force]

TC: $O(N^2)$
SC: $O(1)$

```
for(l=0; l<N; l++){
    for(r=l; r<N; r++){
        sum[l:r]    //use pf sum
    }
}
```

Kadane's algorithm

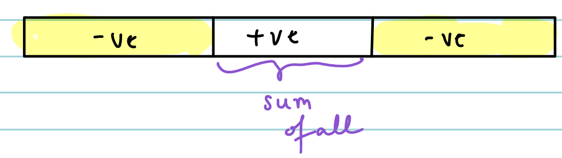
Case 1) If all elements are positive than answer is sum of all

[2, 4, 5, 1, 3]   sum = 15

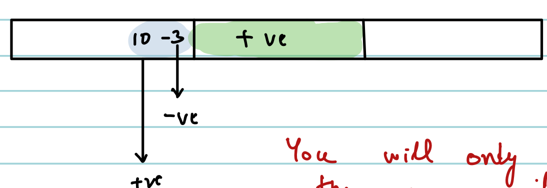
Case 2) If all elements are negative, max will be ans

{-8, -4, -2, -10}   sum = -2

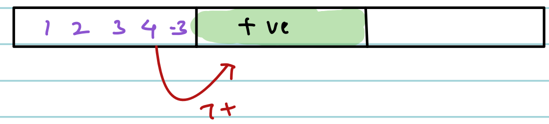
Case 3)

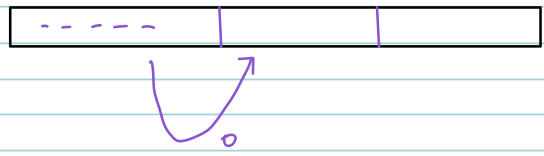


Case 4)



You will only carry forward the sum if sum > 0


1 2 3 4 -3
+ve
7+


0

	0	1	2	3	4	5	6	7	8	9	10	11
	{ 5,	6,	7,	-3,	2,	-10,	-12,	8,	12,	-4,	7,	-2 }
sum = 0	5	11	18	15	17	7	~~-5~~ 0	8	20	16	23	21
maxsum = $-\infty$	5	11	18	18	18	18	18	18	20	20	23	23

	{ -2	-3	-4	-1 }
sum = 0	~~-2~~ 0	~~-3~~ 0	~~-4~~ 0	-1
maxsum = $-\infty$	-2	-2	-2	-1

sum = 0 , maxsum = $-\infty$

```
for (i=0 ; i < N ; i++) {
    sum = sum + A[i]
    if (sum > maxsum) {
        maxsum = sum        # store end ind here
    }
    if (sum < 0) {
        sum = 0             # neglect this
    }
}
```

$\boxed{\text{HW: Max subarray start \& end index}}$

HW

Break (10:30 – 10:40)

Halloween & Christmas are same for soft

31 OCT 25 DEC

$(31)_8 = (25)_{10}$

31: $1 \to 8^0$, $3 \to 8^1$

$1 \times 8^0 + 3 \times 8^1 = 25$

max (subsum(0) , sub(1) ... subar (n-1))

Google

Q4) Max value for the given expression

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

Ex 1)

$$\begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 3, & 7, & 8, & 1, & 2, & 10, & 6, & 9 \end{array}$$

$\{3, 7, 8, 1, 2, 10, 6, 9\}$

$|x| \longrightarrow$ absolute value

$\boxed{abs(x) = max(x, -x)}$

$|-3| \longrightarrow 3$

$abs(-3) = max(-3, 3)$
$= 3$

$abs(-7) = max(-7, -(-7))$

$= max(-7, 7) = 7$

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

$$\begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 3, & 7, & 8, & 1, & 2, & 10, & 6, & 9 \end{array}$$

$\{3, 7, 8, 1, 2, 10, 6, 9\}$

i	j	$\lvert A[i] - A[j]\rvert + \lvert i-j\rvert$
3	5	$\lvert 1-10\rvert + \lvert 3-5\rvert = 9+2 = 11$
7	0	$\lvert 9-3\rvert + \lvert 7-0\rvert = 13$ Ans
3	7	$\lvert 1-9\rvert + \lvert 3-7\rvert = 8+4 = 12$

1) Try all pairs

TC: $O(N^2)$

Q → $\{3, 7, 8, 1, 2, 10, 6, 9\}$ with indices 0 1 2 3 4 5 6 7

Ans = 9

$\max(A[i] - A[j])$

Take dif max − min val

$\max(A[j] - A[i]) \longrightarrow$ ans : 9

Q2) $\left|A[i] - A[j]\right| + (i-j)$

have mod

No mod

$\left|A[i] - A[j]\right| + (i-j)$

$$|x| \longrightarrow \begin{cases} x \\ -x \end{cases}$$

$$\left|A[i] - A[j]\right| \longrightarrow \max \boxed{\begin{cases} A[i] - A[j] + (i-j) \\ A[j] - A[i] + (i-j) \end{cases}}$$

Max
$$\begin{cases} A[i] + i - A[j] - j = \\ A[j] - A[i] + (i-j) = \end{cases} \boxed{\begin{matrix} (A[i] + i) - (A[j] + j) \\ (A[j] - j) - (A[i] - i) \end{matrix}}$$

$A[j] - j - A[i] + i$

$A[i] = A[i] + i$

$$\begin{cases} A[i] - A[j] + (i-j) = \boxed{(A[i]+i) - (A[j]+j)} \rightarrow \\ A[j] - A[i] + (i-j) = (A[j]-j) - (A[i]-i) \end{cases}$$

$$\begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \{3, & 7, & 8, & 1, & 2, & 10, & 6, & 9\} \end{array}$$

$Max(A[j] - A[i]) = max - min$

$max(A'[j] - A'[i])$

$A'[i] = A[i] + i$

$A[i] + i$

A' $\{3 \quad 8 \quad 10 \quad 4 \quad 6 \quad 15 \quad 12 \quad 16\}$

Ans: 13

$A[i] - i$

$\{3 \quad 6 \quad 6 \quad -2 \quad -2 \quad 5 \quad 0 \quad 2\}$

Ans: 8

$A'[i] = A[i] - i$

$max(A'[i] - A'[j])$

$|A[i] - A[j]| + |i-j|$

$A[i] - A[j]$ $\quad$ $i-j$

$A[j] - A[i]$ $\quad$ $j-i$

$A[i] - A[j] + i - j \longrightarrow (A[i] + i) - (A[j] + j)$
$A[i] - A[j] + j - i \longrightarrow (A[i] - i) - (A[j] - j)$
$A[j] - A[i] + i - j \longrightarrow (A[j] - j) - (A[i] - i)$
$A[j] - A[i] + j - i \longrightarrow (A[j] + j) - (A[i] + i)$

$A[i] - A[j]$

Approach 2#

$A[j] - A[i]$

Create 2 arrays

mod +ve $X \longrightarrow X[i] = A[i] + i \longrightarrow \max(X) - \min(X)$

mod -ve $Y \longrightarrow Y[i] = A[i] - i \longrightarrow \max(Y) - \min(Y)$

max

maxv = $-\infty$ minv = ∞

```
for (i=0 ; i<N ; i++) {
    maxv = max(maxv, A[i]+i)
    minv = min(minv, A[i]+i)
}
```

TC: O(N)
SC: O(1)

ans1 = max - minv

maxv = $-\infty$ minv = ∞

```
for (i=0 ; i<N ; i++) {
    maxv = max(maxv, A[i]-i)
    minv = min(minv, A[i]-i)
}
```

ans2 = maxv - minv

return max(ans1, ans2)

0 1 2 3 4 5 6

{ -3 2 4 -1 5 4 3 }

sum → max subarray ending at index i

where index i is always included