Arrays 1 : One Dimensional

Question 1

Given array of N integers , each integers represent the stock price change on that days.

+ive value means price goes up (profit)

-ive value mean price goes down (loss)

Total profit/loss is the total price changes b/w the duration you hold the stock.

find max profit ? (Buy before sell)

A[] = [-20 30 40 -10 50 -100 70]

30 + 40 - 10 + 50 = 110

Short Problem Statement

Given an array, find maximum subarray sum.

A[] = [4 5 2 1 6] ans: 18

A[] = [-4 -3 -6 -9 -2] ans = -2

Bruteforce

∀ subarrays, calculate sum & take max

subarrays = $\frac{N(N+1)}{2}$

```
ans = A[0]
for( i=0 to N-1) {
    for(j=i to N-1) {
        //subarray [i,j]
        sum=0
        for( K=i to j)
            sum += A[K]
        ans = max(ans, sum)
    }
}
```

→ subarray sum

1. prefix sum : pf[j] - pf[i-1]
2. carray forward

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

```
ans = a[0]
for(i=0 to n-1) {
    sum=0
    for(j=i to n-1) {
```

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

```
            sum += a[j]
            ans = max(ans, sum)
        }
    }
```

Observations

1. $\forall i, \; a[i] >= 0 \Rightarrow ans = \sum_{i=0}^{n-1} a[i]$

$A = [4 \;\; 5 \;\; 2 \;\; 1 \;\; 6]$ $ans = 18$

2. $\forall i, \; a[i] < 0 \Rightarrow ans = \forall i \; \max(a[i])$

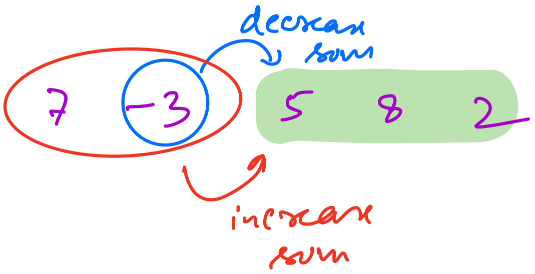
$A = [-4 \;\; -3 \;\; -6 \;\; -9 \;\; -2]$ $ans = -2$

3. A = [-ive -ive +ive +ive +ive -ive -ive]

4. A = [-ive -ive +ive +ive +ive]

A = [+ive +ive +ive -ive -ive]

5. A = [+ive +ive -ive -ive +ive +ive -ive]



A = [-2 3 4 -1 5 -10 -7 2 -5 12]

sum = ~~-2~~ 0 3 7 6 11 1 ~~-6~~ 0 2 ~~-3~~ 0 12

ans = ~~-2~~ ~~3~~ ~~7~~ ~~11~~ 12

A = [-2 3 4 -1 5 -10 7]

sum = ~~-2~~ 0 3 7 6 11 1 8

ans = ~~-2~~ ~~3~~ ~~7~~ 11

Code

Kadane's Algo

```
ans = a[0]
sum = 0          start=0
for (i=0 to N-1) {
    sum += a[i]
```

TC = O(N)

```
        if ( sum > ans) {
              ans = sum
                    L = start , R = i
        }
        if( sum < 0)
              sum = 0     start = i+1

    }
    return ans
```

SC = O(1)

⇒ [L, R] ans subarray

Question 2

Given an array where all elements is zero, ∀i, a[i] = 0.

Return final array after performing multiple queries.

Query : (i, x) → add x to all elements from index i to n-1.

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

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

$$A = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ [\,0 & 0 & 0 & 0 & 0\,] \\ & +3 & +3 & +3 & +3 \\ +2 & +2 & +2 & +2 & +2 \\ & & & & +1 \end{array}$$

Queries

(1,3)

(0,2)

(4,1)

$$A = [\,2 \quad 5 \quad 5 \quad 5 \quad 6\,]$$

Brutforce

$TC = O(Q \times N)$ $\qquad SC = O(1)$

Prefix sum :

$pf[i] = pf[i-1] + a[i]$

$a[i] = a[i-1] + a[i]$

$$A = \begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ [\,0 & \cancel{0} & 0 & \cancel{0} & \cancel{0} & 0 & 0\,] \\ & 3 & & -1 & 2 & & \end{array}$$

$$pf = [\,0 \quad 3 \quad 3 \quad 2 \quad 4 \quad 4 \quad 4\,]$$

Queries

(1,3)

(4,2)

(3,-1)

Code

```
for ( i= 0 to Q.length) {
              i              x
    a[Q[i][0]]  +=  Q[i][1]
                 ↳ what if multiple
                   queries have same index
}

for (i=1 to N-1) {
    a[i] = a[i-1] + a[i]
}
return a
```

TC = $O(Q+N)$

SC = $O(1)$

Question 2 - Part 2

Given an array where all elements is zero, $\forall i, a[i]=0$.

Return final array after performing multiple queries.

Query : (i, j, x) → add x to all elements from index i to j.

$$A = \begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & +3 & +3 & +3 & +3 & & \\ & & & & +2 & +2 & \\ & & & -1 & -1 & -1 & -1 \end{array}$$

$$A = [\,0 \quad 3 \quad 3 \quad 2 \quad 4 \quad 1 \quad -1\,]$$

Queries

(1,4,3)

(4,5,2)

(3,6,-1)

$$A = \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & +3 & +3 & +3 & +3 & & & \\ -1 & -1 & -1 & -1 & -1 & -1 & & \\ & & +4 & & & & & \\ & & & & +3 & +3 & +3 & \end{array}$$

(1,4,3)

(0,5,-1)

(2,2,4)

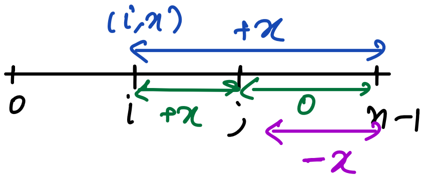
(4,6,3)

$$A = [\,-1 \quad 2 \quad 6 \quad 2 \quad 5 \quad 2 \quad 3 \quad 0\,]$$

$(i, j, x) \Rightarrow [i, j] \quad +x$

$(i, x) \Rightarrow [i, n-1] \quad +x$

$(j+1, -x) \Rightarrow [j+1, n-1] \quad -x$



Code

(i, j, x) : Q[i][0], Q[i][1], Q[i][2]

```
for ( i= 0 to Q.length ) {
                i                  x
    a[ Q[i][0] ]      +=    Q[i][2]
    if( Q[i][1]  <  n-1 ) {
                j                      x
        a[ Q[i][1] +1 ]      -=    Q[i][2]
    }
}
for ( i=1  to  N-1 ) {
    a[i]  = a[i-1]  + a[i]
}
return a
```

TC = O(Q+N)

SC = O(1)

$$A = \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ -1 & 3 & 4 & -4 & 3 & -3 & 1 & -3 \end{array}$$

(1, 4, 3)

(0, 5, -1)

(2, 2, 4)

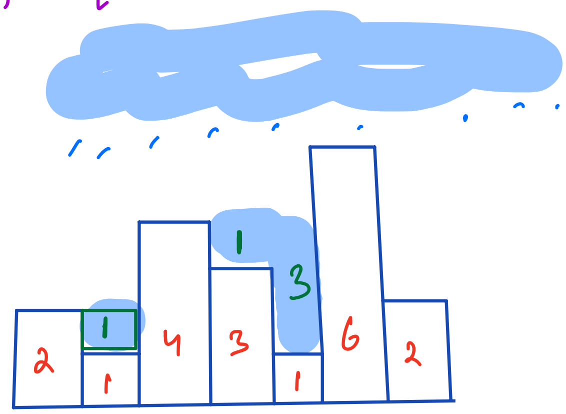
(4, 6, 3)

$$A = [-1 \quad 2 \quad 6 \quad 2 \quad 5 \quad 2 \quad 3 \quad 0]$$

Question 3

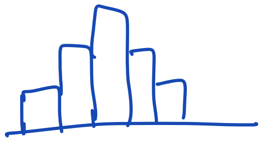
Given N buildings & height of each building. find the rain water trapped b/w buildings.

A= [2 1 4 3 1 6 2]



ans = 1+1+3 = 5

A= [1 2 3 2 1]



ans = 0

Brute force

$\forall i$, find max height in left,
max height in right

take min(left, right) − a[i]

```
ans = 0

for ( i=1   to   n-2 ) {
    maxL  =  max (0, i-1)      //loop O(N)
    maxR  =  max (i+1, n-1)

    water  =  min( maxL, maxR) - a[i]

    if (water > 0) {
        ans  +=  water
    }
}
```

TC = $O(N^2)$

SC = $O(1)$

	0	1	2	3	4	5	6
A = [	2	1	4	3	1	6	2]
maxL		2	2	4	4	4	
maxR		6	6	6	6	2	
min(maxL, maxR)		2	2	4	4	2	
water		1	-2	1	3	-4	

[min(maxL, maxR) - a[i]]

ans = 1+1+3 = 5

Optimize

Create lefMax & right Max array using carry forward.

```
maxL[N] , maxR[N]
maxL[1] = a[0]
for ( i=2 to n-1 ) {
    maxL[i] = max( maxL[i-1], a[i-1] )
}
maxR[n-2] = a[n-1]
for ( i= n-3 to 0 ) {
    maxR[i] = max( maxR[i+1] , a[i+1] )
}
ans = 0
for ( i=1 to n-2 ) {
    water = min( maxL[i], maxR[i] ) - a[i]
    if (water > 0 )
        ans += water
}
```

TC = O(N+N+N)
= O(N)

SC = O(N)