

Arrays I : One Dimensional

Question 1 : Given an integer array, find the max subarray sum.

A = $\begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ [-2 & 3 & 4 & -1 & 5 & -10 & 7] \end{matrix}$

ans = 11

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

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

Brute force → # subarrays, calculate sum & take max

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

$$a_{\min} = \underline{-inf} \text{ or } a[0]$$

for (i=0 to N-1) {

for (j=i to N-1) {

$$\sum v_m = 0$$

for ($K=i$ to j) {

$$\text{sum} += a[k]$$

3

$$ans = \max(ans, sum)$$

subarray sum

$$TC = O(N^3)$$
$$S(=0.1)$$

1. prefix sum $pf[j] = pf[i-1]$

2. Carry forward

$$a_m = a|0\rangle$$

for (i=0 to N-1) {

$$\sum m = 0$$

for (j=i to N-1) {

$sum += a[j] \rightarrow$ carry forward

$$ans = \max(ans, sum)$$

TC: $O(N^2)$

SC: 0017

$$A = \begin{bmatrix} 0 & i & 2 \\ 4 & -2 & 3 \end{bmatrix}$$
$$\text{sum} = 0 - 1$$
$$am = -0.45$$

Observations

$$1. \forall i, a(i) \geq 0 \Rightarrow \text{ans} = \sum_{i=0}^{n-1} a(i)$$

$$A = [4 \ 5 \ 2 \ 1 \ 6] \quad \text{ans} = 18$$

$$2. \forall i, a(i) < 0 \Rightarrow \text{ans} = \forall i \max(a(i))$$

$$A = [-4 \ -3 \ -6 \ -9 \ -2] \quad \text{ans} = -2$$

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

4.

decrease the sum

increase the sum

$$A = \begin{bmatrix} -2 & 3 & 4 & -1 & 5 & -10 & -7 & 2 & -5 & 12 \end{bmatrix}$$

$$\text{sum} = \begin{matrix} -2 & 3 & 7 & 6 & 11 & 1 & -6 & 2 & -3 & 12 \end{matrix}$$

$$\text{ans} = \begin{matrix} -2 & 3 & 7 & 11 \end{matrix}$$

$$12$$

ans = a[0]

sum = 0 start = 0

Kadane's Algo

for (i = 0 to N-1) {

 sum += a[i]

 if (sum > ans) {

 ans = sum

 } L = start, R = i

 if (sum < 0)

 sum = 0

 start = i + 1

⇒ [L - R]

}

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

sum = ~~-2~~⁰ 3 7 6 11 1 8

ans = -2 3 7 11

Question 2

Given an integer array where $\forall i, a[i] = 0$.

Return the final array after performing multiple queries.

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

$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
 $\begin{matrix} +3 & +3 & +3 & +3 & +3 & +3 \\ & +2 & +2 & +2 \\ & & -1 & -1 & -1 & -1 \end{matrix}$
 $\rightarrow 0 \ 3 \ 3 \ 2 \ 4 \ 4 \ 4$



Queries

$(1, 3) \rightarrow TC: O(N)$

$(4, 2)$

$(3, -1)$

$TC: O(B \times N)$

$SL: O(1)$

Prefix sum: $pf[i] = pf[i-1] + a[i]$

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

\rightarrow prefix sum on same array

$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
 $\begin{matrix} 3 & & -1 & 2 \end{matrix}$
 $\rightarrow 0 \ 3 \ 3 \ 2 \ 4 \ 4 \ 4$

Queries

$(1, 3)$

$(4, 2)$

$(3, -1)$

1	3
4	2
3	-1

$B[3][2]$

```

for (i=0 to N-1) { // B[8][2] → B[i][0], B[i][1]
    a[B[i][0]) += B[i][1]
}

```

↳ what if multiple queries have same index

```

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

```

$a = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

$\begin{matrix} +2 \\ =2 \end{matrix}$
 $\begin{matrix} +4 \\ =4 \end{matrix}$

$\begin{matrix} -3 \\ =-1 \end{matrix}$

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

$a = [0 \ -1 \ 0 \ 4]$

↳ after prefix sum

$[0 \ -1 \ -1 \ 3]$

Question 2 - Part 2

Given an integer array where $\forall i, a[i] = 0$.

Return the final array after performing multiple queries.

Query: $(i, j, x) \rightarrow$ add x to all elements from index i to j .

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ +3 & +3 & +3 & +3 & & & \\ & & & +2 & +2 & & \\ & & & -1 & -1 & -1 & -1 \end{bmatrix}$$

→ 0 3 3 2 4 1 -1



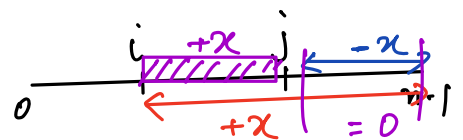
Queries
 (1, 4, 3)
 (4, 5, 2)
 (3, 6, -1)

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ +3 & +3 & +3 & +3 & & & & \\ -1 & -1 & -1 & -1 & -1 & -1 & & \\ & & 4 & & & & & \\ & & & 3 & 3 & 3 & & \end{bmatrix}$$

→ -1 2 6 2 5 2 3 0

Queries
 (1, 4, 3)
 (0, 5, -1)
 (2, 2, 4)
 (4, 6, 3)

$(i, j, x) \Rightarrow i - j + x$
 $(i, x) \Rightarrow i - n - 1 + x$
 $(j+1, -x) \Rightarrow j+1 - n - 1 - x$



for (i=0 to N-1) { // B[i][3] \rightarrow B[i][0], B[i][1], B[i][2]

a[B[i][0]] += B[i][2]

if (B[i][1] < n-1)

a[B[i][1]+1] -= B[i][2]

}

for (i=1 to N-1) {

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

}

return a

A =

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

a = [-1 2 6 2 5 2 3 0]

Queries

(1, 4, 3)

(0, 5, -1)

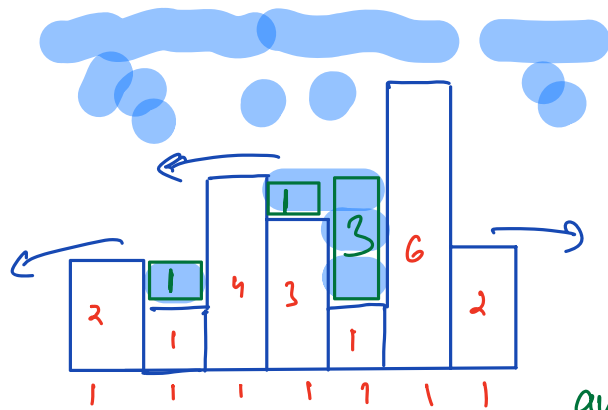
(2, 2, 4)

(4, 6, 3)

Question 3

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

$$A = [2, 1, 4, 3, 1, 6, 2]$$

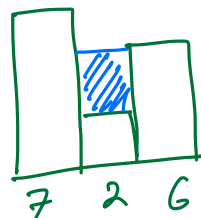


$$am = 1 + 1 + 3 = 5$$

Area

↳ height $\rightarrow a[i]$
↳ base $\rightarrow 1$ & i

$$area = 6 - 2 = 4$$



Area of water above i^{th} building

$$= \min(\text{maxLeft}, \text{maxRight}) - a[i]$$

↳ $O(N)$

$$am = \sum_{i=0}^{n-1} \text{area of water above } i^{th} \text{ building}$$

$$\hookrightarrow N \times N \Rightarrow TC: O(N^2) \quad SC: O(1)$$

ans = 0

for $i = 1$ to $N-2$ {

maxL = max(0... i-1) // loop $O(N)$ \rightarrow prefix max

maxR = max(i+1... N-1) // loop $O(N)$ \rightarrow suffix max

$$\text{water} = \min(\text{maxL}, \text{maxR}) - a[i]$$

if (water < 0) water = 0

ans += water

}

A = $\begin{bmatrix} 2 & 1 & 4 & 3 & 1 & 6 & 2 \end{bmatrix}$

maxL = $\begin{bmatrix} 2 & 2 & 4 & 4 & 4 & 6 & 6 \end{bmatrix}$

maxR = $\begin{bmatrix} 6 & 6 & 6 & 6 & 6 & 6 & 2 \end{bmatrix}$

min(
maxL,
maxR)

water = $\begin{bmatrix} 0 & 1 & 0 & 1 & 3 & 0 & 0 \end{bmatrix} \Rightarrow \text{ans} = 5$

maxL[i] =
max(maxL[i-1], a[i])

maxR[i] =
max(maxR[i+1], a[i])

Code

maxL[0] = a[0]

for (i = 1 to n-1)

maxL[i] = max(maxL[i-1], a[i])

}

maxR[n-1] = a[n-1]

for (i = n-2 to 0)

maxR[i] = max(maxR[i+1], a[i])

}

TC: O(N)

SC: O(N)

ans = 0

for (i = 1 to n-2) {

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

}

return ans

We can remove maxL[n] by doing carry forward
for prefix max.

maxR[n-1] = a[n-1]

for (i = n-2 to 0) {

maxR[i] = max(maxR[i+1], a[i])

}

ans = 0, maxL = a[0]

SC = O(N)

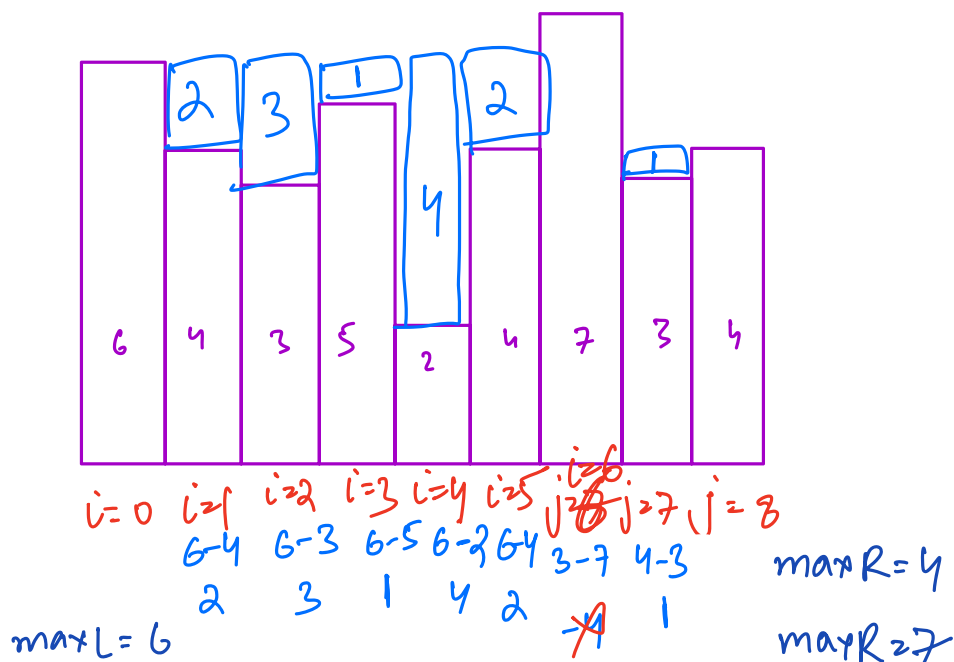
for (i = 1 to n-2) {

maxL = max(maxL, a[i])

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

}

$A = [6, 4, 3, 5, 2, 4, 7, 3, 4]$



$$2 + 3 + 1 + 4 + 2 + 1 = 13$$

Code

$i=0, j=n-1$

$ans=0$

$lmax = a[0], rmax = a[n-1]$

while ($i < j$) {

if ($lmax < rmax$) {

$i++$

$water = lmax - a[i]$

$lmax = \max(lmax, a[i])$

}

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

else {

j--

water = rmax - a[i]

rmax = max(rmax, a[i])

}

if (water > 0)

ans += water

}

return ans