

P1.1 Given an integer array A, ($A[i]$ is initially 0 for all i) return

prequel the final array after processing multiple queries:

Query(i, x) → add x to all the numbers from index i to $n-1$

ex $a[] = \{$

0	0	0	0	0	0	0
0	+3	+3	+3	+3	+3	+3
0	0	0	0	+2	+2	+2
0	0	0	+1	+1	+1	+1

$\}$

Query

$i \rightarrow x$

(1, 3)

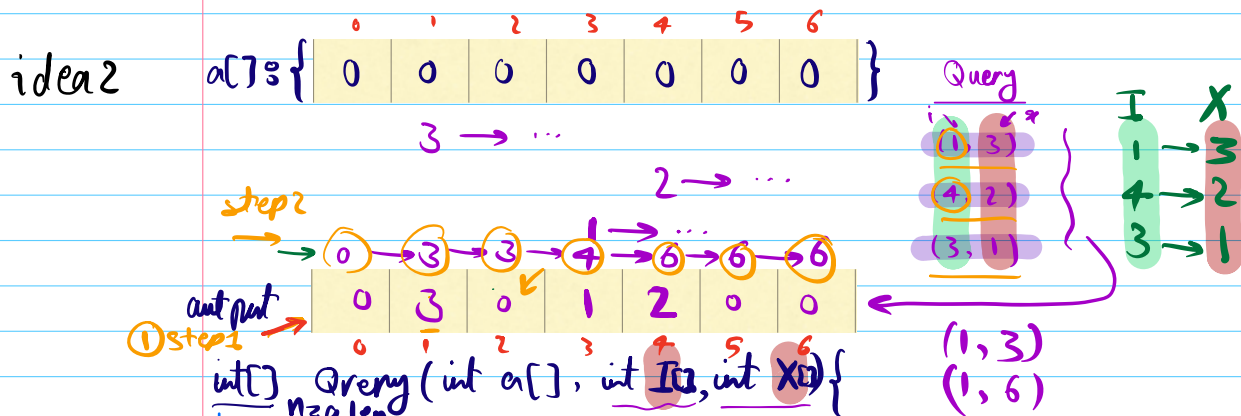
(4, 2)

(3, 1)

output 0 3 3 4 6 6 6

idea1 for each Q, start from i

TC: $O(Q \times N)$ SC: $O(1)$



TC:

$O(Q + N)$

SC:

$O(1)$

$int[]$ Query($int a[], int I[], int X[]$)

$int q = I.length // or X.length$

① for ($i = 0; i < q; i++$) {

index = $I[i]$; $x = X[i]$;
 $a[index] += x$;

② for ($i = 1; i < n; i++$) {
 $a[i] += a[i-1]$;
}

ret a;

P1.2 Given an integer array A, ($A[i]$ is initially 0 for all i) return the final array after processing multiple queries:

Query(i, j, x) \rightarrow add x to all the numbers from index i to j

ex $a[] = \{$

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

$\}$

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

Queries

(1, 3, 2)

(2, 5, 3)

(5, 6, -1)

output

0	2	5	5	3	2	-1
---	---	---	---	---	---	----

$(i, j, x) \rightarrow$
 $\begin{cases} (i, x) \text{ ①} \\ (j+1, -x) \text{ ②} \end{cases}$

0	1	2	3	4	5	6

TC: $O(2Q + N) = O(Q + N)$

SC: $O(1)$

P2.1 Given an integer array A , find max value of $f(i, j)$

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

ex $a[] = \{1, 3, -2\}$

idea1
 $\text{for}(i = 0 \rightarrow n-1)$
 $\text{for}(j = 0 \rightarrow n-1)$
 $\max(A[i] - A[j])$

TC: $O(n^2)$

i	j	$A[i] - A[j] \rightarrow f(i, j)$
0	0	0
0	1	-2
0	2	+3
1	0	2
1	1	0
1	2	5 \rightarrow ans
2	0	-3
2	1	-5
2	2	0

idea2
 $f(i, j) = A[i] - A[j]$
 $\text{ret } \max - \min$

$\max = \text{int_min}; \quad \min = \text{int_max}$

```
for i = 0  $\rightarrow$  n-1 {
    max = math.max(max, a[i])
    min = math.min(min, a[i])
}
ret max - min
```

TC: $O(n)$

SC: $O(1)$

P2.2 Given an integer array A, find max value of $f(i, j)$

$$f(i, j) = |A[i] - A[j]| + |i - j| \quad \text{for all } i, j$$

ex $a[] = \{ \overset{0}{1} \ \overset{1}{3} \ \overset{2}{-2} \}$

$$|x| \geq 0$$

brute force
T.C. $O(n^2)$

i	j	① $ A[i] - A[j] $	② $ i - j $	$f(i, j)$
0	0	0	0	0
0	1	+2	1	3
0	2	+3	2	5
1	0	2	1	3
1	1	0	0	0
1	2	5	1	6
2	0	+3	2	5
2	1	+5	1	6
2	2	0	0	0

↗
↖
cng

$$\{ \overset{0}{0} \ 100 \ 1 \ 1 \ 1 \ 1 \ \overset{7}{7} \}$$

observations

① $i = j \rightarrow f(i, j) = 0$ \times $O(n^2)$

② $\underline{f(i, j)} = \underline{f(j, i)} \leftarrow$

$$|x| = \begin{cases} x & x \geq 0 \\ -x & x < 0 \end{cases}$$

optimize:
$$f(i, j) = |A[i] - A[j]| + (i - j) \quad i > j$$

$A[i] \geq A[j]$

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

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

$X_i \quad X_j$

$A[k] + k = X_k$

max
 $f(i, j) \leftarrow \text{MAX of } X_i - \text{MIN of } X_j$
 ①

$A[i] \leq A[j]$

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

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

$Y_j \quad Y_i$

$A[k] - k = Y_k$

max
 $f(i, j) = Y_j - Y_i$
 ②

$\downarrow \text{max} \quad \downarrow \text{min}$

$$\text{Ans} = \max(X_{\max} - X_{\min}, Y_{\max} - Y_{\min})$$

$\max X = \text{int_min}; \min X = \text{int_max}; \max Y = \text{int_min}; \min Y = \text{int_max}$

for ($k=0; k < n; k++$) {

$X = a[k] + k; \quad Y = a[k] - k$

$\max X = \text{math.max}(\max X, X);$

$\min X = \text{math.min}(\min X, X);$

$\max Y = \text{math.max}(\max Y, Y);$

$\min Y = \text{math.min}(\min Y, Y);$

}

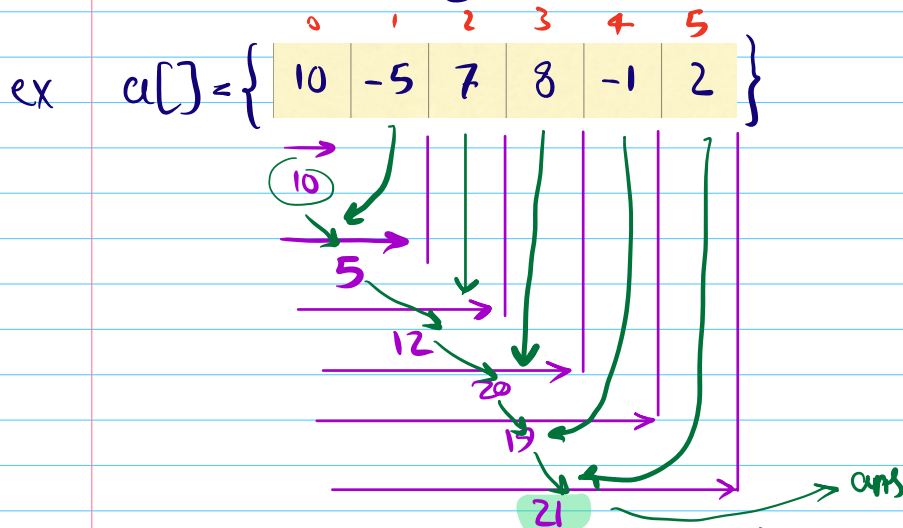
$\text{ret } \text{math.max}(\max X - \min X, \max Y - \min Y);$

$Tc: O(n)$

$Sc: O(1)$

break?

P3.1 Given an integer array $a[]$, find the **max subarray sum** for subarrays starting from **index 0**.



```
int maxSubarraySum1(int a[])
```

```
    n = a.len; sum = 0; ans = INT_MIN;
```

```
    for(i = 0; i < n; i++) { // carry forward {-1, -2}
```

```
        sum += a[i];
        ans = math.max(ans, sum);
    }
```

```
    return ans;
```

```
}
```

Tc:

$O(n)$

Sc:

$O(1)$

P3.2 Given an integer array a , find the max subarray sum for all the subarrays.

ex $a[] = \{ 10, -5, 7, 8, -1, 2 \}$

observations

① $a[i] \geq 0$ for all i $\sum a[i] \rightarrow \text{ans}$

② $a[i]$ for all i $a[i] < 0 \rightarrow \text{max of } a[i] \leftarrow \text{ans}$

③

10, -5, 7, 8, 1, 2

18 ← sum

black box

sum < 0

 $O(n)$

Kaden's

Algorithm

[illegible]



```
int maxSubarraySum2(int a[])
```

```
n = a.len
```

```
ans = a[0]
```

```
sum = 0
```

```
for(i = 0; i < n; i++) {
```

```
    sum += a[i]
```

```
    ans = math.max(ans, sum);
```

```
    if (sum < 0) sum = 0;
```

```
}
```

```
return ans;
```

```
}
```

	0	1	2	3
A =	-3	-8	-1	-5
sum	0	0	0	0
ans	-3	-3	-1	-1