

Contest Discussion

a) Little pony & Man element

b) Vowels in Range

c) Manimium Positivity ✓

Note: Stay back and have a discussion
with me? o Solved

Maximum positivity

Given an $arr[N]$ return max size subarray with all $ele \geq 0$
if there are more than one such subarray, return one having smallest start index.

Ex1: $arr[10] =$

0	1	2	3	4	5	6	7	8	9
-3	1	3	0	-6	2	3	1	4	-4

 $ans = \{2, 3, 1, 4\}$
 \downarrow $len = 3$ \downarrow $len = 4$

Ex2: $arr[10] =$

0	1	2	3	4	5	6	7	8	9
-3	1	3	0	-6	2	3	1	-8	4

 $ans = \{1, 3, 0\}$
 \downarrow $len = 3$ \downarrow $len = 3$ \downarrow $len = 1$

Ex3: $arr[10] =$

0	1	2	3	4	5	6	7	8	9
-3	1	3	0	-6	2	3	1	4	4

 $ans = \{2, 3, 1, 4, 4\}$
 \downarrow $len = 3$ \downarrow $len = 5$

Quiz1: start = 2, len = 5 subarray: $\{2, 3, 4, 5, 6\}$ $e = s + l - 1 = 6$

Quiz2: start = 4, len = 4 subarray: $\{4, 5, 6, 7\}$ $e = s + l - 1 = 4 + 4 = 7$

Quiz3: start = 5, len = 5 subarray: $\{5, 6, 7, 8, 9\}$ $e = s + l - 1 = 5 + 5 - 1 = 9$

obs: $\left. \begin{array}{l} \text{Start of Subarray} = s \\ \text{length of Subarray} = l \\ \text{End of Subarray} = e = s + l - 1 \end{array} \right\} \text{Subarray: } \{s, s + l - 1\} = \cancel{s + l - 1} - \cancel{s + 1} + 1 = l$

Note: If we have start index of Subarray & len of subarray = Get End of Subarray.
And we can also get subarray.

Ex1: arr[10] =

0	1	2	3	4	5	6	7	8	9
-3	1	3	0	-6	2	3	1	4	-4
→ *	✓	✓	✓	*	✓	✓	✓	✓	*

as = 5

al = 4

{s=1}
{l=3}

{s=5}
{l=4}

// final as = 5, al = 4, ae = 8

// Iterate on as to ae & get Subarray.

Ex2:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
-2	3	-1	4	5	0	-3	-4	1	3	5	6	7	-8	2	4	8	
✓	✓	*	✓	✓	✓	*	*	✓	✓	✓	✓	✓	*	✓	✓	✓	end

as = -1 = initialization of ans variable

al = 0

i

0 : +ve, S = 0

1, 2, l = 2, update as = 0, al = 2

3 : +ve S = 3

3, 4, 5, 6, l = 3, update as = 3, al = 3

7 : -ve : goto next index

8 : +ve S = 8

8, 9, 10, 11, 12, 13, l = 5, update as = 8, al = 5

14 : +ve S = 14

14, 15, 16, 17, l = 3, no update as = 8, al = 5

ArrayList<Integer> maxLen(int arr[]) { TC: O(N)

We iterate on arr[]
only once, so TC: O(N)

int n = arr.length;

int as = -1; ad = 0;

int i = 0;

while(i < n) { → ①

// Say at ith index, when can we start subarray from ith index.

if (arr[i] < 0) { // goto next index i = i + 1

else { // start from ith index

int s = i; int l = 0;

while (i < n && arr[i] >= 0) → ②

l = l + 1;

i = i + 1

{ Before we check arr[i] >= 0 }
{ check, if i is < n }

// Once loop is done we have s & l

if (l > ad) { // curr len > ans len

as = s; ad = l

int ae = as + ad - 1 //

// We know as ... ae, Can we get subarray

ArrayList<Integer> ans = new ArrayList<Integer> <>();

for (int i = as; i <= ae; i++) {

ans.add(arr[i])

}

return ans;

	0	1	2	3	4	5	6	7	8	9	10	11	12
arr:	-2	3	1	-4	-5	0	3	4	-1	3	5	6	7

$as = -1, al = 0, i = 0$

Outerloop

$i = 0$ $arr[i] < 0 : i = i + 1 = i = 1$

$i = 1$ $arr[i] \geq 0 : \text{inner loop}$

$s = 1, l = 0$

$arr[1] \geq 0 : l = l + 1, i = i + 1; i = 2$

$arr[2] \geq 0; l = l + 1, i = i + 1; i = 3$

$arr[3] \geq 0$

$s = 1, l = 2 : \text{update } as = 1, al = 2$

$i = 3$ $arr[i] < 0 : i = i + 1; i = 4$

$i = 4$ $arr[i] < 0 : i = i + 1; i = 5$

$i = 5$ $arr[i] \geq 0 : \text{inner loop}$

$s = 5, l = 0$

$arr[5] \geq 0 : l = l + 1, i = i + 1, i = 6$

$arr[6] \geq 0; l = l + 1, i = i + 1, i = 7$

$arr[7] \geq 0; l = l + 1, i = i + 1, i = 8$

$arr[8] \geq 0$

$s = 5, l = 3 : \text{update } as = 5, al = 3$

$i = 8$ $arr[i] < 0 : i = i + 1; i = 9$

$i = 9$ $arr[i] \geq 0 : \text{inner loop}$

$s = 9, l = 0$

$arr[9] \geq 0; l = l + 1, i = i + 1, i = 10$

$arr[10] \geq 0, l = l + 1, i = i + 1, i = 11$

$arr[11] \geq 0, l = l + 1, i = i + 1, i = 12$

$arr[12] \geq 0, l = l + 1, i = i + 1, i = 13$

$arr[i] \geq 0 : \text{end of array}$

28) Little pony & maximum element:

Given $arr[N]$, in 1 step we can set any $ele = -1$

Min operations required to make max of $arr[]$ as B

Ex: $arr[6] = \{4, 2, 6, 1, 2, 5\}$ $ans = 3$
 $B = 2 \rightarrow$ $\begin{matrix} 4 & 2 & 6 & 1 & 2 & 5 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ -1 & 2 & -1 & 1 & 2 & -1 \end{matrix}$ $ele > B, \text{ updated them.}$

Ex2: $arr[7] = \{4, 7, 6, 2, 1, 3, 8\}$ $ans = 4$
 $B = 3 \rightarrow$ $\begin{matrix} 4 & 7 & 6 & 2 & 1 & 3 & 8 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ -1 & -1 & -1 & 2 & 1 & 3 & -1 \end{matrix}$ $ele > B, \text{ update them}$

Ex3: $arr[6] = \{6, 7, 3, 5, 9, 2\}$ // array does not contain 4
 $B = 4 \rightarrow$ $\begin{matrix} 6 & 7 & 3 & 5 & 9 & 2 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ -1 & -1 & 3 & -1 & -1 & 2 \end{matrix}$ // max can never be 4
 // return -1

int minOpe(int arr[], int B) { Tc: $O(N)$ Sc: $O(1)$

int c = 0

boolean ispresent = false;

int n = arr.length

for (int i = 0; i < n; i++)

{ if (arr[i] == B) { ispresent = true; }

if (arr[i] > B) { c++; // update arr[i] = -1; }

if (ispresent == false) { return -1; }

else { return c; }

3Q) Even Numbers in Range

Given String: N & Queries: Q

For each query i, j calculate no. of vowels in given range

String S = 0 1 2 3 4 5 6 7 8 9
 a b c a e i g h o a

Q: 3

i j // vowels

2 6 : 3

4 8 : 3

2 8 : 4

Idea1: For every query iterate on query and get count. TC: $Q \times N$

Idea2: Optimize using Psum[]

String S = 0 1 2 3 4 5 6 7 8 9
 a b c a e i g h o a
 ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
int A[10] = 1 0 0 1 1 1 0 0 1 1
 ↗ ↗ ↗ ↗ ↗ ↗ ↗ ↗ ↗
int pSum[10] = 1 1 1 2 3 4 4 4 5 6

i j : pSum[j] - pSum[i-1]

2 6 : pSum[6] - pSum[1] = 3

4 8 : pSum[8] - pSum[3] = 3

2 8 : pSum[8] - pSum[1] = 4

0 3 : pSum[3] = 2

```
void RangeCount(char ch[N], int Q, int L[Q], int R[Q]) {
```

Tc: $O(N+Q)$

Sc: $O(N)$

```
    int ar[N];
```

```
    for (i = 0; i < N; i++) { // Step 1: arr[] → Tc:  $O(N)$ 
```

```
        if (ch[i] == 'a' || ch[i] == 'e' || ch[i] == 'i' || ch[i] == 'o' || ch[i] == 'u') {
```

```
            ar[i] = 1
```

```
        }
```

```
        else { ar[i] = 0 }
```

```
    }
```

```
    int pSum[N] // Step 2: pSum we perform on modified array → Tc:  $O(N)$ 
```

```
    pSum[0] = ar[0]
```

In modified arr[], each element is 0/1

```
    for (i = 1; i < N; i++) {
```

According to constraints $1 \leq N \leq 10^5$

```
        pSum[i] = pSum[i-1] + ar[i] // Max sum we can get =  $10^5$  → int ✓
```

```
    }
```

```
    for (i = 0; i < Q; i++) { → Tc:  $Q * O(1)$ 
```

```
        // ith Query: L[i] ... R[i]
```

```
        int s = L[i], e = R[i]
```

```
        if (s == 0) { // no: even of [0..e] / print (pSum[e]) }
```

```
        else { // no: even of s..e / print (pSum[e] - pSum[s-1]) }
```

```
    }
```

```
}
```