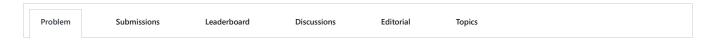


The Maximum Subarray





Given an array $A = \{a_1, a_2, \dots, a_N\}$ of N elements, find the maximum possible sum of a

- 1. Contiguous subarray
- 2. Non-contiguous (not necessarily contiguous) subarray.

Empty subarrays/subsequences should not be considered.

Input Format

First line of the input has an integer T. T cases follow.

Each test case begins with an integer N. In the next line, N integers follow representing the elements of array A.

Constraints:

- $1 \le T \le 10$
- $1 \leq N \leq 10^5$
- $\bullet \ -10^4 \leq a_i \leq 10^4$

The subarray and subsequences you consider should have at least one element.

Output Format

Two, space separated, integers denoting the maximum contiguous and non-contiguous subarray. At least one integer should be selected and put into the subarrays (this may be required in cases where all elements are negative).

Sample Input

```
2
4
1 2 3 4
6
2 -1 2 3 4 -5
```

Sample Output

```
10 10
10 11
```

Explanation

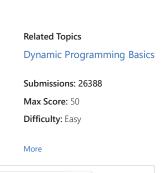
In the first case:

The max sum for both contiguous and non-contiguous elements is the sum of ALL the elements (as they are all positive).

In the second case

[2 -1 2 3 4] --> This forms the contiguous sub-array with the maximum sum.

For the max sum of a not-necessarily-contiguous group of elements, simply add all the positive elements.



Current Buffer (saved locally, editable) 🐉 🧿

1 of 2 01-05-2016 06:51 PM

```
1 ▼import java.io.*;
 2 import java.util.*;
 3 import java.text.*;
   import java.math.*;
 5 import java.util.regex.*;
7 ▼public class Solution {
8
        public static void main(String[] args) {
           /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
10
            Scanner read = new Scanner(System.in);
11
12
            int t = read.nextInt();
13 ▼
            for(int i=0;i<t;i++){
                int n = read.nextInt();
14
15
                int[] arr = new int[n];
16 ▼
                for(int j=0;j<n;j++){
17
                   arr[j]=read.nextInt();
18
19
                int maxSumSoFar=Integer.MIN VALUE;
                int maxEnding = 0;
20
21
                int nonContSum=0;
22 ▼
                for(int x: arr){
                    maxEnding = Math.max(x, maxEnding+x);
2.3
                    maxSumSoFar = Math.max(maxEnding, maxSumSoFar);
24
25
26
27
                    System.out.print(maxSumSoFar);
28
29
30
                //2) For max non-continuous sub array, order doesn't matter
31
                Arrays.sort(arr);
32
                int sum = 0;
33
34
                //{\rm if} there is none positive value in entire array
35 ▼
                if(arr[arr.length-1] \le 0){
36
                    sum = arr[arr.length - 1];
37
                    //System.out.println("neg case: "+sum);
38
                //accumulate all positive values in entire array
39 ▼
40 ▼
                    for(int x : arr) {
41
                         if(x > 0)
42
                           sum += x;
43
44
45
                System.out.println(" " + sum);
47 }
                                                                                                                       Line: 1 Col: 1
                    Test against custom input
1 Upload Code as File
                                                                                                              Run Code
                                                                                                                         Submit Code
                                                     Copyright © 2016 HackerRank. All Rights Reserved
```

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature

2 of 2 01-05-2016 06:51 PM