

[Practice](#)[Compete](#)[Find Jobs](#)[Rank](#)[Leaderboard](#)

sandeepraikar ▾

[All Domains](#) > [Algorithms](#) > [Dynamic Programming](#) > The Maximum SubarrayBadge Progress [\(Details\)](#)

Points: 921.00 Rank: 14874

The Maximum Subarray

 by [sh4d0wkn1ght](#)

Problem

Submissions

Leaderboard

Discussions

Editorial

Topics

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:

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

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.

Related Topics

[Dynamic Programming Basics](#)

Submissions: 26388

Max Score: 50

Difficulty: Easy

[More](#)

Current Buffer (saved locally, editable)

Java 7 ▾



```
1 import java.io.*;
2 import java.util.*;
3 import java.text.*;
4 import java.math.*;
5 import java.util.regex.*;
6
7 public class Solution {
8
9     public static void main(String[] args) {
10         /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
11         Scanner read = new Scanner(System.in);
12         int t = read.nextInt();
13         for(int i=0;i<t;i++){
14             int n = read.nextInt();
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;
20             int maxEnding = 0;
21             int nonContSum=0;
22             for(int x: arr){
23                 maxEnding = Math.max(x, maxEnding+x);
24                 maxSumSoFar = Math.max(maxEnding, maxSumSoFar);
25             }
26
27             System.out.print(maxSumSoFar);
28
29             //2) For max non-continuous sub array, order doesn't matter
30             Arrays.sort(arr);
31             int sum = 0;
32
33             //if there is none positive value in entire array
34             if(arr[arr.length-1] <= 0){
35                 sum = arr[arr.length - 1];
36                 //System.out.println("neg case: "+sum);
37             //accumulate all positive values in entire array
38             }else{
39                 for(int x : arr){
40                     if(x > 0)
41                         sum += x;
42                 }
43             }
44             System.out.println(" " + sum);
45         }
46     }
47 }
```

Line: 1 Col: 1

[Upload Code as File](#) ☐ Test against custom input

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](#)