The Maximum Subarray ☆

148/563 challenges solved

Rank: 3540 | Points: 3606.67

Problem Solving ******

Problem Submissions Leaderboard Editorial Topics

We define subsequence as any subset of an array. We define a subarray as a contiguous subsequence in an array.

Given an array, find the maximum possible sum among:

1. all nonempty subarrays.
2. all nonempty subsequences.

Print the two values as space-separated integers on one line.

Note that empty subarrays/subsequences should not be considered.

For example, given an array $arr = [-1, 2, 3, -4, 5, 10]$, the maximum subarray sum is comprised of element inidices $[1 - 5]$ and the sum is $2 + 3 + -4 + 5 + 10 = 16$. The maximum subsequence sum is comprised of element indices $[1, 2, 4, 5]$ and the sum is $2 + 3 + 5 + 10 = 20$.

Function Description

Complete the maxSubarray function in the editor below. It should return an array of two integers: the maximum subarray sum and the maximum subsequence sum of arr.

maxSubarray has the following parameter(s):

- arr: an array of integers

Input Format

The first line of input contains a single integer t, the number of test cases.

The first line of each test case contains a single integer n.
The second line contains n space-separated integers $arr[i]$ where $0 \leq i < n$.

Constraints

- $1 \leq t \leq 10$
- $1 \leq n \leq 10^5$
- $-10^4 \leq arr[i] \leq 10^4$

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

Output Format

Print two space-separated integers denoting the maximum sums of nonempty subarrays and nonempty subsequences, respectively.

Sample Input 0

```
2
4
1 2 3 4
6
```

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

Sample Output 0

```
10 10
10 11
```

Explanation 0

In the first case: The maximum sum for both types of subsequences is just the sum of all the elements since they are all positive.

In the second case: The subarray $[2, -1, 2, 3, 4]$ is the subarray with the maximum sum, and $[2, 2, 3, 4]$ is the subsequence with the maximum sum.

Sample Input 1

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

Sample Output 1

```
-1 -1
```

Explanation 1

Since all of the numbers are negative, both the maximum subarray and maximum subsequence sums are made up of one element, -1.

Current Buffer (saved locally, editable)

Java 8

```
1  import java.io.*;
2  import java.math.*;
3  import java.security.*;
4  import java.text.*;
5  import java.util.*;
6  import java.util.concurrent.*;
7  import java.util.regex.*;
8
9  public class Solution {
10
11     // Complete the maxSubarray function below.
12     static int[] maxSubarray(int[] arr) {
13
14
15     }
```

```
16
17      private static final Scanner scanner = new Scanner(System.in);
18
19      public static void main(String[] args) throws IOException {
20          BufferedWriter bufferedWriter = new BufferedWriter(new
    FileWriter(System.getenv("OUTPUT_PATH")));
21
22          int t = scanner.nextInt();
23          scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");
24
25          for (int tItr = 0; tItr < t; tItr++) {
26              int n = scanner.nextInt();
27              scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");
28
29              int[] arr = new int[n];
30
31              String[] arrItems = scanner.nextLine().split(" ");
32              scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");
33
34              for (int i = 0; i < n; i++) {
35                  int arrItem = Integer.parseInt(arrItems[i]);
36                  arr[i] = arrItem;
37              }
38
39              int[] result = maxSubarray(arr);
40
41              for (int i = 0; i < result.length; i++) {
42                  bufferedWriter.write(String.valueOf(result[i]));
43
44                  if (i != result.length - 1) {
45                      bufferedWriter.write(" ");
46                  }
47              }
48
49              bufferedWriter.newLine();
50          }
51
52          bufferedWriter.close();
53
54          scanner.close();
55      }
56  }
57
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code