Practice Compete Jobs Rank Leaderboard

Dashboard > Algorithms > Constructive Algorithms > Two Subarrays

Badge Progress (Details)

Points: 126 Rank: 270332

Two Subarrays

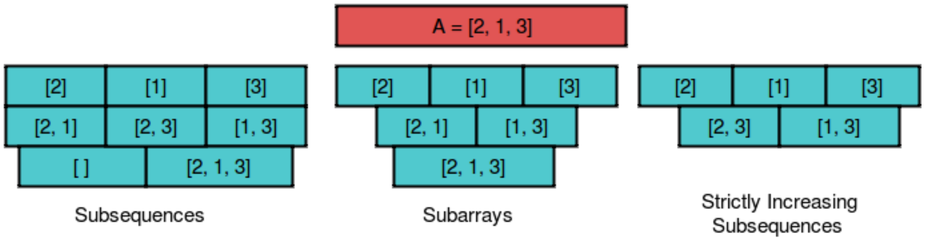
by allleksssa

Problem Submissions Leaderboard Discussions Editorial

Consider an array, $A = a_0, a_1, \ldots, a_{n-1}$, of n integers. We define the following terms:

- **Subsequence**
  A subsequence of A is an array that's derived by removing zero or more elements from A without changing the order of the remaining elements. Note that a subsequence may have zero elements, and this is called *the empty subsequence*.
- **Strictly Increasing Subsequence**
  A non-empty subsequence is *strictly increasing* if every element of the subsequence is larger than the previous element.
- **Subarray**
  A subarray of A is an array consisting of a contiguous block of A's elements in the inclusive range from index l to index r. Any subarray of A can be denoted by $A[l,r] = a_l, a_{l+1}, \ldots, a_r$.

The diagram below shows all possible subsequences and subarrays of $A = [2,1,3]$:



We define the following functions:

- $sum(l,r) = a_l + a_{l+1} + \ldots + a_r$
- $inc(l,r)$ = the maximum sum of some *strictly increasing subsequence* in subarray $A[l,r]$
- $f(l,r) = sum(l,r) - inc(l,r)$

We define the *goodness*, g, of array A to be:

$$g = max\ f(l,r) \text{ for } 0 \le l \le r < n$$

In other words, g is the maximum possible value of $f(l,r)$ for all possible subarrays of array A.

Let m be the length of the smallest subarray such that $f(l,r) = g$. Given A, find the value of g as well as the number of subarrays such that $r - l + 1 = m$ and $f(l,r) = g$, then print these respective answers as space-separated integers on a single line.

Input Format

The first line contains an integer, n, denoting number of elements in array A.
The second line contains n space-separated integers describing the respective values of $a_0, a_1, \ldots, a_{n-1}$.

Constraints

- $1 \le n \le 2 \cdot 10^5$

- $-40 \leq a_i \leq 40$

Subtasks

For the 20% of the maximum score:

- $1 \leq n \leq 2000$
- $-10 \leq a_i \leq 10$

For the 60% of the maximum score:

- $1 \leq n \leq 10^5$
- $-12 \leq a_i \leq 12$

Output Format

Print two space-seperated integers describing the respective values of g and the number of subarrays satisfying $r - l + 1 = m$ and $f(l, r) = g$.

Sample Input 0

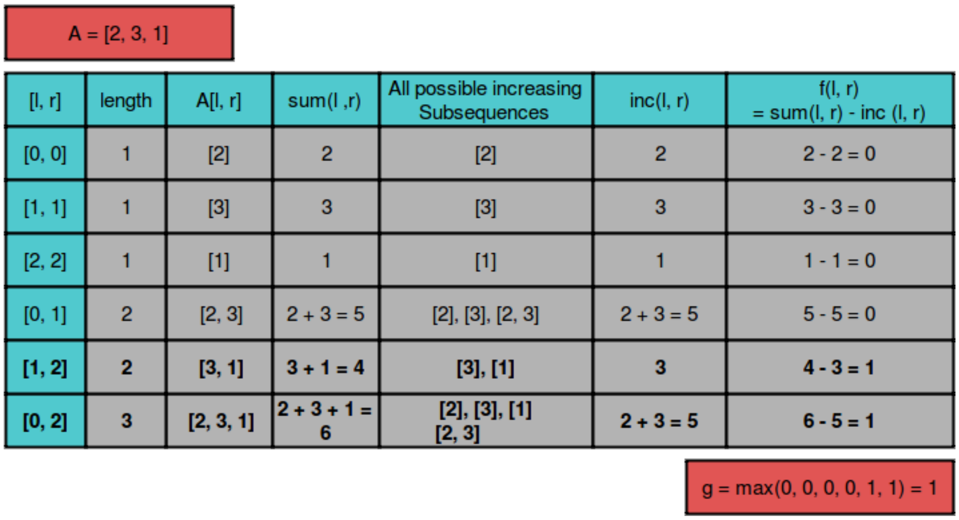
```
3
2 3 1
```

Sample Output 0

```
1 1
```

Explanation 0

The figure below shows how to calculate g:



A = [2, 3, 1]

[l, r]	length	A[l, r]	sum(l ,r)	All possible increasing Subsequences	inc(l, r)	f(l, r) = sum(l, r) - inc (l, r)
[0, 0]	1	[2]	2	[2]	2	2 - 2 = 0
[1, 1]	1	[3]	3	[3]	3	3 - 3 = 0
[2, 2]	1	[1]	1	[1]	1	1 - 1 = 0
[0, 1]	2	[2, 3]	2 + 3 = 5	[2], [3], [2, 3]	2 + 3 = 5	5 - 5 = 0
[1, 2]	**2**	**[3, 1]**	**3 + 1 = 4**	**[3], [1]**	**3**	**4 - 3 = 1**
[0, 2]	**3**	**[2, 3, 1]**	**2 + 3 + 1 = 6**	**[2], [3], [1] [2, 3]**	**2 + 3 = 5**	**6 - 5 = 1**

g = max(0, 0, 0, 0, 1, 1) = 1

m is the length of the smallest subarray satisfying $f(l, r)$. From the table, we can see that $m = 2$. There is only one subarray of length 2 such that $f(l, r) = g = 1$.

Submissions:67

Max Score:70

Difficulty: Expert

Rate This Challenge:

☆ ☆ ☆ ☆ ☆

More

Current Buffer (saved locally, editable)

Java 7

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int n = in.nextInt();
        int[] a = new int[n];
        for(int a_i=0; a_i < n; a_i++){
            a[a_i] = in.nextInt();
        }
        // your code goes here
    }
}

```

Line: 1 Col: 1

Upload Code as File

□ Test against custom input

Run Code

Submit Code

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