Subsequence Weighting

by HackerRank

Problem Submissions Leaderboard Discussions

A subsequence of a sequence is a sequence which is obtained by deleting zero or more elements from the sequence.

You are given a sequence `A` in which every element is a pair of integers i.e `A` = $[(a_1, w_1), (a_2, w_2), ..., (a_N, w_N)]$.

For a subseqence `B` = $[(b_1, v_1), (b_2, v_2), ..., (b_M, v_M)]$ of the given sequence :

- We call it increasing if for every i ($1 <= i < M$) , $b_i < b_{i+1}$.
- $Weight(B) = v_1 + v_2 + ... + v_M$.

Task:
Given a sequence, output the maximum weight formed by an increasing subsequence.

Input:
The first line of input contains a single integer T. T test-cases follow. The first line of each test-case contains an integer N. The next line contains $a_1, a_2, ..., a_N$ separated by a single space. The next line contains $w_1, w_2, ..., w_N$ separated by a single space.

Output:
For each test-case output a single integer: The maximum weight of increasing subsequences of the given sequence.

Constraints:
$1 <= T <= 5$
$1 <= N <= 150000$
$1 <= a_i <= 10^9$, where $i \in [1..N]$
$1 <= w_i <= 10^9$, where $i \in [1..N]$

Sample Input:

```
2
4
1 2 3 4
10 20 30 40
8
1 2 3 4 1 2 3 4
10 20 30 40 15 15 15 50
```

Sample Output:

```
100
110
```

Explanation:
In the first sequence, the maximum size increasing subsequence is 4, and there's only one of them. We choose `B = [(1, 10), (2, 20), (3, 30), (4, 40)]`, and we have `Weight(B) = 100`.

In the second sequence, the maximum size increasing subsequence is still 4, but there are now 5 possible subsequences:

```
1 2 3 4
10 20 30 40

1 2 3 4
10 20 30 50

1 2 3 4
10 20 15 50

1 2 3 4
10 15 15 50

1 2 3 4
15 15 15 50
```

Of those, the one with the greatest weight is `B = [(1, 10), (2, 20), (3, 30), (4, 50)]`, with `Weight(B) = 110`.

Please note that this is not the maximum weight generated from picking the highest value element of each index. That value, 115, comes from [(1, 15), (2, 20), (3, 30), (4, 50)], which is not a valid subsequence because it cannot be created by only deleting elements in the original sequence.

Submissions:1254

Max Score:60
Difficulty: Advanced

Rate This Challenge:

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     }
12 }
```

Line: 1 Col: 1

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