



G. Mukhammadali and the Smooth Array

time limit per test: 1 second
memory limit per test: 256 megabytes



Muhammadali has an integer array a_1, \dots, a_n . He can change (replace) any subset of positions; changing position i costs c_i and replaces a_i with any integer of his choice. The positions that he does not change must retain their original values.

After all changes, we call an index i ($1 \leq i < n$) a *drop* if the final value at position i is strictly greater than the final value at position $i + 1$. Muhammadali wants the final array to contain no *drops*.

Find the minimum cost of changes required to ensure that there are no *drops* in the array.

有道 翻译



Muhammadali有一个整数数组 a_1, \dots, a_n 。他可以改变（替换）任何位置子集；改变位置 i 花费 c_i ，并将 a_i 替换为他选择的任何整数。他没有改变的立场必须保持其原有的价值。

在所有更改之后，如果位置 i 的最终值严格大于位置 $i + 1$ 的最终值，则将索引 i ($1 \leq i < n$) 称为*drop*。Muhammadali希望最后的数组不包含任何水滴。

找出确保数组中没有下降所需的最小更改成本。



Input

The first line contains an integer t ($1 \leq t \leq 5000$) — the number of test cases.

Each test case consists of three lines:

The first line contains a single integer n ($1 \leq n \leq 8000$) — the length of the arrays.

The second line contains n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — the elements of the array.

The third line contains n integers c_1, c_2, \dots, c_n ($1 \leq c_i \leq 10^9$) — the costs of changes.

It is guaranteed that the sum of n across all test cases does not exceed 8000.

有道 翻译



输入** **

第一行包含一个整数 t ($1 \leq t \leq 5000$) — 测试用例的数量。

每个测试用例由三行组成：

第一行包含一个整数 n ($1 \leq n \leq 8000$) —— 数组的长度。

第二行包含 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) —— 数组的元素。

第三行包含 n 整数 c_1, c_2, \dots, c_n ($1 \leq c_i \leq 10^9$) - 更改的成本。

可以保证所有测试用例的 n 之和不超过 8000。



Codeforces Round 1062 (Div. 4)

比赛进行中

01:45:53

Contestant



→ 提交?

语言: GNU G++17 7.3.0

选择文件: 选择文件 未选择文件

提交

Output

↑ For each test case, output a single integer — the minimum possible total cost required to eliminate all *drops*.

有道 翻译



** **输出

对于每个测试用例，输出一个整数——消除所有掉落所需的最小可能的总成本。

Example

input

Copy

```
10
1
10
5
4
1 2 2 3
5 6 7 8
4
4 3 2 1
1 1 1 1
3
3 1 2
100 1 1
5
5 5 5 5 5
10 1 10 1 10
5
1 3 2 2 4
100 1 1 1 100
6
10 9 8 7 6 5
1 100 1 100 1 100
5
100 1 100 100 100
1 100 1 1 1
4
2 1 2 1
5 4 3 2
7
1 5 3 4 2 6 7
10 1 10 1 10 1 10
```

output

Copy

```
9
0
0
3
2
0
1
203
1
6
11
```



Note

In the first and second examples, the array already has no *drops*, so no changes are needed.

In the third example, one of the optimal arrays is: $[2, 3, 5, 6]$; to achieve this, all elements except the second need to be replaced, so the answer is $c_1 + c_3 + c_4 = 3$.

有道 翻译



注意

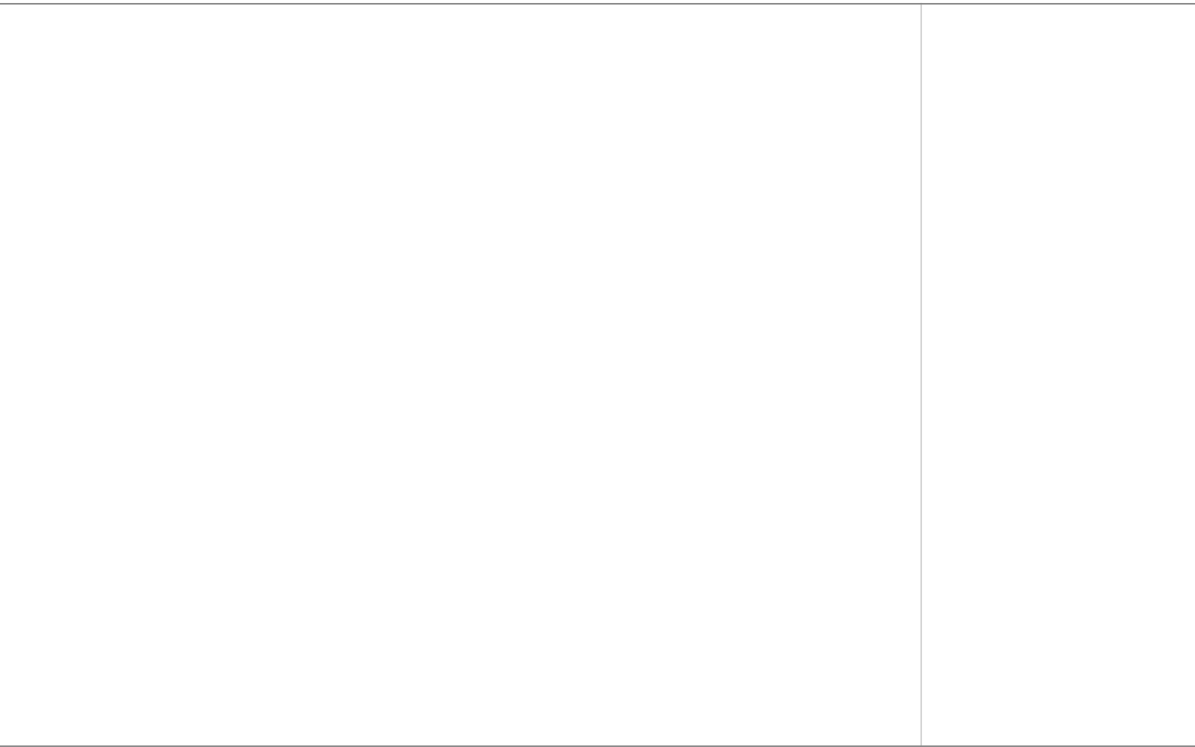
在第一个和第二个示例中，数组已经没有掉落，因此不需要更改。

在第三个例子中，最优数组之一是： $[2, 3, 5, 6]$ ；要做到这一点，除了第二个元素之外的所有元素都需要被替换，所以答案是 $c_1 + c_3 + c_4 = 3$ 。

GNU G++17 7.3.0



1



► 自定义测试数据(自动保存)



[Codeforces](#) (c) Copyright 2010-2025 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Oct/28/2025 23:03:57^{UTC+8} (k2).
Desktop version, switch to [mobile version](#).
[Privacy Policy](#) | [Terms and Conditions](#)

Supported by



ITMO