Task Planning

≜ locked

Submissions: 32

Difficulty: Medium

Rate This Challenge:

Max Score: 10

More

Problem Submissions Leaderboard Discussions

An employee is trying to plan his tasks. There is a **deadline** for each task, along with a **penalty** charged if the task was submitted after the deadline. Each task takes one unit of time only.

Given a set of tasks, we want to make a **timetable** detailing the order in which to submit these tasks. The first task in the timetable begins at time 0 and finishes at time 1, the second task begins at time 1 and finishes at time 2, and so on.

For the given set of tasks, we want to find a timetable that **minimizes the total penalty** acquired for missing deadlines. You are required to answer the question above using a **greedy algorithm**.

Input Format

- The first line will contain **N**, which is the **number of tasks**.
- The second line will contain N space-separated integers representing the **deadlines** *d0 d1 d2 ... dN-1* of each task.
- The third line will contain N space-separated integers representing the **penalties** *p0 p1 p2 ... pN-1* incurred. If the task ai is not finished by the time di of each task, we incur a penalty pi.

Constraints

- 1 ≤ N ≤ 12000
- 1 ≤ di ≤ N
- 0 ≤ pi ≤ 10^8

Output Format

One line containing the **total penalty** for an optimal timetable.

Sample Input 0

```
7
6 1 4 3 2 4 4
10 30 50 40 60 20 70
```

Sample Output 0

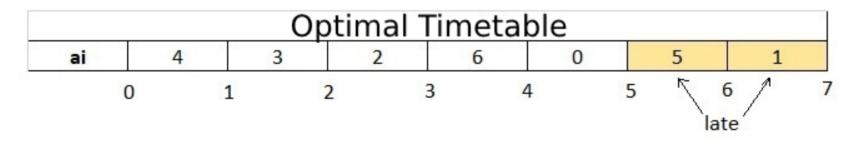
50

Explanation 0

The optimal solution (the solution having the minimum total penalty) has a total penalty = 50. This can be accomplished by the timetable: 4 3 2 6 0 5 1 (other timetables can also be optimal with total penalty = 50).

Tasks 1 and 5 will be late (submitted after deadline d1 = 1 and d5 = 4) and they will be penalized by p1 = 30 and p5 = 20 which is 50.

| Tasks | | | | | | | |
|-------|----|----|----|----|----|----|----|
| ai | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| di | 6 | 1 | 4 | 3 | 2 | 4 | 4 |
| pi | 10 | 30 | 50 | 40 | 60 | 20 | 70 |



```
C++20
 1 ▼ #include <cmath>
   #include <cstdio>
   #include <vector>
   #include <iostream>
   #include <algorithm>
   using namespace std;
8
9 vint main() {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT */
10 ▼
11
        return 0;
12
13
                                                                                               Line: 1 Col: 1
```

<u>Upload Code as File</u> Test against custom input

Run Code

Submit Code