

Programmieraufgabe P1.

/ 20 P

Passwort für Einschreibung: asymptotic**Einreichung:** <https://judge.inf.ethz.ch/team/websubmit.php?cid=28784&problem=ALGOTowe>**ALGO Tower**

You are given a collection of $n \leq 5000$ of *ALGO bricks*, indexed from 1 to n (see example below). The i -th ALGO brick is a rectangular cuboid having a $w_i \times d_i$ base, with $w_i, d_i \in \{1, 2, \dots, 10000\}$, and height $h_i \in \{0, 1, \dots, 1000\}$.

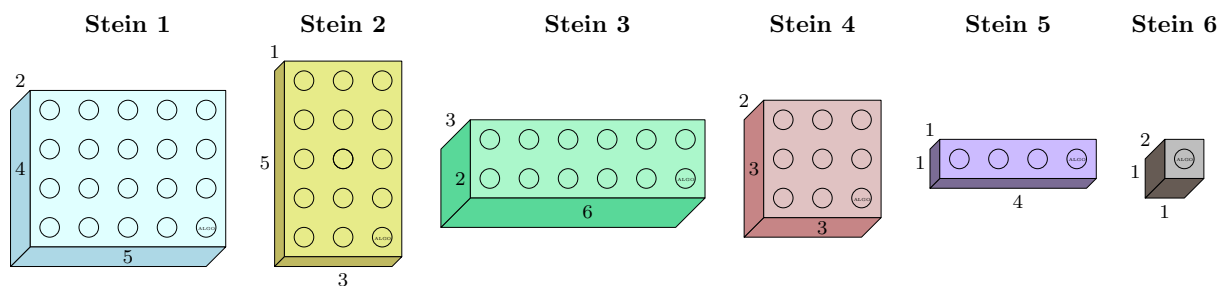
We want to build a tower from these blocks that is *as tall as possible*. To build such a tower, we can pile bricks on top of each other, subject to having *no overhang*. Formally, the i -th brick can be placed on top of the j -th brick if and only if at least one of the following two conditions holds:

- $w_i \leq w_j$ and $d_i \leq d_j$; or
- $d_i \leq w_j$ and $w_i \leq d_j$;

Intuitively, the second condition corresponds to *rotating* the i -th block by 90 degrees.

An ALGO tower is a list $\langle t_1, t_2, \dots, t_k \rangle$ of distinct indices such that, if $k \geq i > j \geq 1$, then the t_i -th brick can be placed on top of the t_j -th brick. The height of T is the sum of the heights of the involved bricks, i.e., $\sum_{i=1}^k h_{t_i}$.

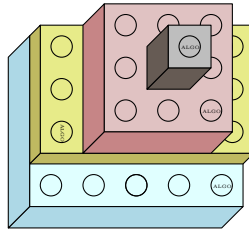
Your task is to compute the height of the tallest *ALGO tower* that can be built with the available bricks. You can assume that all base areas are distinct and that the bricks are initially sorted by decreasing base area, i.e., if $1 \leq i < j \leq n$, then $w_i \cdot d_i > w_j \cdot d_j$.

Beispiel

Consider the $n = 6$ ALGO bricks shown above, and defined by the following dimensions:

Brick 1 $w_1 = 5, d_1 = 4, h_1 = 2$;**Brick 4** $w_4 = 3, d_4 = 3, h_4 = 2$;**Brick 2** $w_2 = 3, d_2 = 5, h_2 = 1$;**Brick 5** $w_5 = 1, d_5 = 4, h_5 = 1$;**Brick 3** $w_3 = 6, d_3 = 2, h_3 = 3$;**Brick 6** $w_6 = 1, d_6 = 1, h_6 = 2$;

The tallest ALGO tower that can be built with these bricks is $\langle 1, 2, 4, 6 \rangle$, as shown by the picture below. It has height $h_1 + h_2 + h_4 + h_6 = 2 + 1 + 2 + 2 = 7$.



Notice that Brick 2 (in yellow) was *rotated* in order to place it on top of Brick 1 (in blue).

Anforderung

Overall, you can obtain a maximum of 20 judge points for this programming task. To get full points your program should run in time $O(n^2)$, with reasonable hidden constants. Slower solutions might get partial points:

Subtask 1: You can obtain up to 8 points by correctly solving instances that (i) involve at most $n \leq 200$ bricks, and (ii) are guaranteed to have an optimal solution in which no brick needs to be rotated (with respect to the input orientation).

Subtask 2: You can obtain up to 8 additional points (i.e., 16 points in total) if your program correctly solves instances with $n \leq 200$. Here, optimal solutions can also involve rotated bricks.

Greedy solutions will likely not obtain any points (see, e.g., the second test case in the example input).

Instruktionen

For this exercise, we provide a program template as an Eclipse project in your workspace. The template already contains the code needed to read the input and write the output.

The project also contains data for your local testing and a `Judge.java` program that runs your `Main.java` on all the local tests – just open and run `Judge.java` in the project. The local test data are different from the data that are used in the online judge.

Submit only your `Main.java`.

(Continues on the next page)

Die Ein- und Ausgabe werden von der Vorlage verarbeitet – Sie sollten den Rest dieses Texts nicht benötigen.

Eingabe Die erste Zeile der Eingabe enthält einzig die Anzahl der Tests.

Each test case consists of 4 lines: The first line contains only the integer n ; The second line contains the integers w_1, \dots, w_n , separated by a space; The third line contains the integers d_1, \dots, d_n , separated by a space; The fourth line contains the integers h_1, \dots, h_n , separated by a space.

Ausgabe For each test case, output – on a separate line – the height of the tallest ALGO tower that can be built with the available ALGO bricks.

Beispiel-Eingabe (the first test case corresponds to the example above):

```
2
6
5 3 6 3 4 1
4 5 2 3 1 1
2 1 3 2 1 2
3
10 13 11
4 3 2
8 4 6
```

Beispiel-Ausgabe:

```
7
10
```

Platz für Ihre Notizen. Diese werden nicht bewertet. Nur was auf dem Judge eingereicht wird zählt für diese Aufgabe.

