Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Department Informatik
Markus Püschel
Peter Widmayer
Thomas Tschager
Tobias Pröger
Tomáš Gavenčiak

3rd November 2016

Datenstrukturen & Algorithmen Exercise Sheet P7 AS 16

Hand-in: Before Thursday, 10th November 2016 10:00 via the online judge (source code only).

Exercise P7.1 *Presents and ribbons.*

Halloween is over and the North Pole is slowly preparing for Christmas. There are already many gifts waiting to be wrapped in ribbons and you are the one who has to do it this time. You want to save yourself some work with tying new ribbons (and the planet resources, of course) by reusing as many already-tied ribbons from last year. However the ribbons must fit the boxes perfectly and you have to figure out how many of them you can reuse.

Every already-tied ribbon is a loop with given length in cm. Every present is a cube[1] and someone has measured the area of one face of every box in cm^2. A ribbon of length l fits a box with area a only when $\sqrt{a} = l/4$, that is when l is exactly the circumference of a square with area a. For example, a ribbon of length $40\,cm$ fits a cube of area of one side $100\,cm^2$.

To make your task easier, the $0 \leq m$ ribbons are already sorted by their lengths $l_0 \leq l_1 \leq \cdots \leq l_{m-1}$, and the $0 \leq n$ boxes are sorted by their side areas $a_0 \leq a_1 \leq \cdots \leq a_{n-1}$. All the lengths and areas are natural numbers and no rounding is allowed. We recommend to avoid floating numbers and only use integers.

Given the ribbons and boxes and their data, find the maximal number of ribbon-box pairs that you can reuse this year. There may be more boxes of the same size and more ribbons of the same length, and you should pair as many of them as possible. One ribbon can of course only be used for one box, and one box needs just one ribbon.

Input The input consists of four lines. The first line contains just the integer m. The second line contains m integers l_0 to l_{m-1} (ribbon lengths) separated by spaces and in ascending order. The third line contains just the integer n. The fourth line contains n integers a_0 to a_{n-1} (box side areas) separated by spaces and in ascending order.

Output The output should contain one integer number, the number of reusable ribbons.

Grading You will get 1 bonus point for every 100 judge points, rounded down. Your may get up to 200 judge points. The program should be reasonably efficient and work in $O(n)$ time or similar to get full points.

Submit your `Main.java` at `https://judge.inf.ethz.ch/team/websubmit.php?cid=18985&problem=DA_P7.1`, enroll password is "`quicksort`".

[1] That is with all side lengths equal.

Examples

Input:

```
8
4 5 12 12 14 32 32 33
10
1 4 5 9 9 9 49 55 64 65
```

Output (matching 4 to 1, 12 to 9, 12 to 9, 32 to 64):

```
4
```

Notes For this exercise, we provide a program template as an Eclipse project archive on the lecture website, which will load the input for you. The archive also contains more tests for you convenience – you can copy&paste the data into your running program.

We recommend to avoid binary search – there is a much better suited sorting-related technique for this task.