

Code Jam 2013 - Round 1B

Osmos

Problem

Armin is playing Osmos, a physics-based puzzle game developed by Hemisphere Games. In this game, he plays a "mote", moving around and absorbing smaller motes.

A "mote" in English is a small particle. In this game, it's a thing that absorbs (or is absorbed by) other things! The game in this problem has a similar idea to Osmos, but does not assume you have played the game.

When Armin's mote absorbs a smaller mote, his mote becomes bigger by the smaller mote's size. Now that it's bigger, it might be able to absorb even more motes. For example: suppose Armin's mote has size 10, and there are other motes of sizes 9, 13 and 19. At the start, Armin's mote can only absorb the mote of size 9. When it absorbs that, it will have size 19. Then it can only absorb the mote of size 13. When it absorbs that, it'll have size 32. Now Armin's mote can absorb the last mote.

Note that Armin's mote can absorb another mote if and only if the other mote is *smaller*. If the other mote is the same size as his, his mote can't absorb it.

You are responsible for the program that creates motes for Armin to absorb. The program has already created some motes, of various sizes, and has created Armin's mote. Unfortunately, given his mote's size and the list of other motes, it's possible that there's no way for Armin's mote to absorb them all.

You want to fix that. There are two kinds of operations you can perform, in any order, any number of times: you can add a mote of any positive integer size to the game, or you can remove any one of the existing motes. What is the minimum number of times you can perform those operations in order to make it possible for Armin's mote to absorb every other mote?

For example, suppose Armin's mote is of size 10 and the other motes are of sizes [9, 20, 25, 100]. This game isn't currently solvable, but by adding a mote of size 3 and removing the mote of size 100, you can make it solvable in only 2 operations. The answer here is 2.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. The first line of each test case gives the size of Armin's mote, **A**, and the number of other motes, **N**. The second line contains the **N** sizes of the other motes. All the mote sizes given will be integers.

Output

For each test case, output one line containing "Case #x: y", where x is the case number (starting from 1) and y is the minimum number of operations needed to make the game solvable.

Limits

Time limit: 30 seconds per test set.

Memory limit: 1GB.

$1 \leq T \leq 100$.

Small dataset (Test set 1 - Visible)

$1 \leq A \leq 100$.

$1 \leq \text{all mote sizes} \leq 100$.

$1 \leq N \leq 10$.

Large dataset (Test set 2 - Hidden)

$1 \leq A \leq 10^6$.

$1 \leq \text{all mote sizes} \leq 10^6$.

$1 \leq N \leq 100$.

Sample

Sample Input

```
4
2 2
2 1
2 4
2 1 1 6
10 4
25 20 9 100
1 4
1 1 1 1
```

Sample Output

```
Case #1: 0
Case #2: 1
Case #3: 2
Case #4: 4
```

Notes

Although the size of motes is limited in the input files, Armin's mote may grow larger than the provided limits by absorbing other motes.

Osmos was created by Hemisphere Games. Hemisphere Games does not endorse and has no involvement with Google Code Jam.