Problem Buyer

Time Limit: 8000/4000 MS (Java/Others) Memory Limit: 32768/32768 K (Java/Others)

Total Submission(s): 218 Accepted Submission(s): 66

Problem Description

TopSetter is an organization that creates problems. They've prepared N problems with estimated difficulty score in range [A_i, B_i]. TopHoster would like to host a contest consisting of M problems.
The i^{th} problem should be of difficulty score C_i. The i^{th} problem from TopSetter can be used in the contest if and only if its estimated difficulty score range $[A_i, B_i]$ covers the difficulty score c of its target problem in the contest, i.e. $A_i \leq c \leq B_i$. Hosting a contest with M problems needs tohave M distinct problems which satisfy the required difficulty scores for each problem.
Unfortunately, TopSetter doesn't provide a service to buy specific problems. You can only request a problem set containing K problems and they will give you K distinct problems from all the N problems, but you don't know which problems will be given.
As TopSetter is the only problem provider for TopHoster, TopHoster would like to know the least number K of problems they need to buy to make sure they can host a contest.

Input

The first line of the input gives the number of test cases, T. T test cases follow. Each test case starts with 2 integers, N and M. Then N lines follow, each line consists of 2 integers representing the difficulty score range of the i^{th} problem, $A_i and B_i$. The last line of each test case consists of M integers representing the target difficulty scores of the M problems C_i .

Output

For each test case, output one line containing "Case #x: y", where x is the test case number (starting from 1) and y is the least number of problems which the TopHoster needs to buy.
Output "IMPOSSIBLE!" if it's impossible.

limits

- $1 \leq T \leq 100$.
- $1 \leq N, M \leq 10^5$.
- $1 \leq A_i \leq B_i \leq 10^9$.
- $1 \leq C_i \leq 10^9$.

Sample Input

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

Sample Output

```
Case #1: 2
```

```
Case #2: 2
Case #3: IMPOSSIBLE!
```

Source

2016 CCPC-Final

Recommend

jiangzijing2015 | We have carefully selected several similar problems for you: 6216 6215 6214 6213 6212