

Library Hell

Lea is running out of storage space on the system. She knows there are some large software packages installed that she never uses (like Qt), but she remembers that removing them can break some packages she needs for filling the order for more storage (like LibreOffice).

Lea has numbered all the packages on the system and exported the complete list of package dependencies. After preparing the list of packages to remove and the list of packages to keep, she needs to check that the plan is safe. Check that all packages to keep continue working, i.e., that they, their dependencies, dependencies of their dependencies etc. are not removed. While Lea herself can never make such a mistake, some of her friends might borrow the tool and specify the same package to keep and to remove, or no packages in one (or both) of the lists.

Input

The first line of the input contains an integer t . t test cases follow, each of them separated by a blank line.

Each test case starts with a line containing four space-separated integers N , K , R , and D . Here N is the total number of packages, K is the number of packages to keep working, R is the number of packages to remove, and D is the number of package dependency records.

A line containing K space-separated integers k_1, \dots, k_K follows. The packages with numbers k_1, \dots, k_K and all their indirect (transitive) dependencies have to be preserved. The next lines contains R space-separated integers r_1, \dots, r_R , denoting the list of packages planned for removal.

Afterwards D lines follow describing the dependencies. Each of these lines contains two space-separated integers u_i and d_i meaning that the package number u depends on the package number d .

Output

For each test case, output one line containing “Case # i : ok” if the dependency requirements are satisfied, or “Case # i : conflict” otherwise.

Constraints

- $1 \leq t \leq 100$
- $1 \leq N \leq 10000$
- $0 \leq K \leq N$
- $0 \leq R \leq N$
- $0 \leq D \leq 200000$
- $1 \leq k_i \leq N$
- $1 \leq r_i \leq N$
- $1 \leq u_i \leq N$
- $1 \leq d_i \leq N$

Additionally, the total number of lines in a single test input shall not exceed 2 millions.

Sample Input 1

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

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

Sample Output 1

```
Case #1: ok
Case #2: conflict
```

Sample Input 2

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

5 0 1 1

1
2 3
```

Sample Output 2

```
Case #1: ok
Case #2: ok
```