

6302 Star Travel

In year 3001, star travel is realized by a device called StarGate. With the StarGate, travelers can hop from a planet to a planet via hyperspace. However, due to some physical constraints on hyperspace, it is not always possible to establish StarGates between every two planets. In addition, the transporting is directional; that is, it is possible to transport from one StarGate to another but not vice versa. So, traveling to a planet is often done by hopping a series of StarGates geometrically.

However, in that era, human colonies are divided into several federations. A planet only belongs to a federation but a traveler can own the citizenship from more than one federation. A planet can operate more than one StarGate to other planets. A traveler can transport between two StarGates free, i.e, without tickets, if he owns the citizenship from the federations that control the two StarGates. Now, a traveler wants to travel from one planet to another. Please write a program to book a sequence of tickets for him, excluding the free tickets. Assume tickets are of the same price. Please plan a tour that has the minimum number of tickets.

Input

The test data begins with an integer N , which is the number of test cases. Each test case begins with an integer P , which is the number of planets. Following P is the P lines of planet data. The planet data begins with the ID of a planet. The planet ID is a non-negative integer ranges from 0 to 5000. The second item of planet data is the federation ID. Federation ID is a capital letter ranges from 'A' to 'Z'. Next is a number of outgoing StarGates on that planet, which is a non-negative integer S ranges from 0 to 100. When S is zero, no outgoing StarGates are deployed on that planet. Each outgoing StarGate is described by a ticket name Tx and destination planet ID that can be reached by this StarGate, where x is a non-negative integer from 0 to 99999999, which is not necessarily monotonically increasing in the test data. The ticket names never duplicate.

Following the planet data is the traveler data. The data begins with an integer T , which is the number of tours to be planned for the planets. Following is T lines of tour and traveler data. Each data contains the source planet ID and destination planet ID that he wants to travel. Following the two IDs is a string that contains the citizenship of federations of the traveler. For example, if a traveler owns the citizenship from federation A, B, and C. The string will be "ABC". Every traveler owns at least a citizenship from a federation.

Output

For each test case, please output the sequence of ticket names for each tour, excluding the free tickets. Each ticket name is separated by a space. If there are more than one solution, please only output the sequence that begins with the smaller ticket names. For example, suppose two sequences (T1 T3 T4) and (T6 T3 T8) are both valid answers. You only need to output (T1 T3 T4) because $T1 < T6$. If ticket names are equal, the rule applies consecutively until a difference is made.

Sample Input

```
2
5
0 A 2 T0 1 T2 2
1 A 1 T1 2
2 A 1 T3 3
```

```
3 B 1 T4 4
4 B 1 T5 1
1
0 4 A
6
0 A 2 T0 1 T2 2
1 A 1 T1 2
2 A 2 T3 3 T6 5
3 B 1 T4 4
4 B 1 T5 1
5 A 1 T7 4
2
0 4 A
5 3 A
```

Sample Output

```
T3 T4
T7
T7 T5 T3
```