7-4 Dijkstra Sequence (30 分)

Dijkstra's algorithm is one of the very famous greedy algorithms. It is used for solving the single source shortest path problem which gives the shortest paths from one particular source vertex to all the other vertices of the given graph. It was conceived by computer scientist Edsger W. Dijkstra in 1956 and published three years later.

In this algorithm, a set contains vertices included in shortest path tree is maintained. During each step, we find one vertex which is not yet included and has a minimum distance from the source, and collect it into the set. Hence step by step an ordered sequence of vertices, let's call it **Dijkstra sequence**, is generated by Dijkstra's algorithm.

On the other hand, for a given graph, there could be more than one Dijkstra sequence. For example, both { 5, 1, 3, 4, 2 } and { 5, 3, 1, 2, 4 } are Dijkstra sequences for the graph, where 5 is the source. Your job is to check whether a given sequence is Dijkstra sequence or not.

Input Specification:

Each input file contains one test case. For each case, the first line contains two positive integers N_v ($\leq 10^3$) and N_e ($\leq 10^5$), which are the total numbers of vertices and edges, respectively. Hence the vertices are numbered from 1 to N_v.

Then N_e lines follow, each describes an edge by giving the indices of the vertices at the two ends, followed by a positive integer weight (≤ 100) of the edge. It is guaranteed that the given graph is connected.

Finally the number of queries, K, is given as a positive integer no larger than 100, followed by K lines of sequences, each contains a permutationof the N_v vertices. It is assumed that the first vertex is the source for each sequence.

All the inputs in a line are separated by a space.

Output Specification:

For each of the K sequences, print in a line `Yes` if it is a Dijkstra sequence, or `No` if not.

Sample Input:

```
5 7
1 2 2
1 5 1
2 3 1
2 4 1
2 5 2
3 5 1
3 4 1
4
5 1 3 4 2
5 3 1 2 4
2 3 4 5 1
3 2 1 5 4
```

Sample Output:

```
Yes
Yes
Yes
No
```

编译器 (33)   C++ (g++) ▼   帮助

1

提交