13143   Dijkstractions

Astrijdk Geders has two passions: her cat and Dijkstra's algorithm. She loves walking with her cat in a cage through the shortest paths of her town.

She also spends hours and hours studying properties and variations of Dijkstra's algorithm. For example, she has studied how to adapt the algorithm when the graph is directed and acyclic, or what to do when the cost of a path is given by the product of the edges, instead of the sum.

Now she is studying an even more crazily complex variation: what happens when the cost of a path with edges $(a, b, c, d, ...)$ is given by a successive exponentiation

$$a^{b^{c^{d^{\cdots}}}}$$

Given a directed acyclic graph weighted with integer numbers (from 1 to 10^6), find the longest path between two given nodes. In this problem, the length of a path is defined as the successive exponentiation of the cost of the edges. That is, if the path goes through the edges $(a, b, c, ...)$, in that order, then the cost of the path is $a^{b^{c^{\cdots}}}$. So, for example, if the path has only one edge of cost a, the length of the path is a; if it has edges (a, b), the length is a^b; and so on.

Please, observe that: a^{b^c} is the same as: $a^{(b^c)}$, which is not the same as $(a^b)^c$.

Input

The input consists of a series of test cases. The first line contains a number that indicates the number of test cases.

Each case begins with two numbers, N and A, indicating the number of nodes and edges of the graph, respectively. Nodes are numbered from 1 to N, and N is not greater than 10000. The following line contains two different numbers, S and E, the start and the end of the path, respectively. Then, there are A lines, each of them with three numbers: V W C, indicating an edge from V to W with cost C. Remember that the graph is directed an acyclic.

Output

For each test case, the program has to output one line. This line is the sequence of nodes of the longest path, starting with S and ending with E, separated by spaces.

If there is no path, the output will be an empty line. If there are many optimal solutions, you have to output the one that goes through the node with lowest number first (that is, the solution with the lowest lexicographical order).

Sample Input

```
3
4 4
1 4
1 2 4
2 4 1
1 3 2
3 4 8
4 4
4 1
```

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

Sample Output

```
1 3 4

4 2 5 3 1
```