Problem2 Finding Tour Path

Suyeon who is a student at Kaist is planning to travel to Europe after this semester. She found a lot of famous cities such as Berlin, Zurich, Paris, and so on, in Europe and want to go everywhere on travel. However, there is no enough time to travel everywhere because she has to come back Kaist for her research as soon as possible. So, she are thinking about a shortest tour path between cities that she want to go. She considers two cases to find a tour path. First is a path which only has a source city and destination city. Second is a path containing stopover cities as well as a source city and destination city.

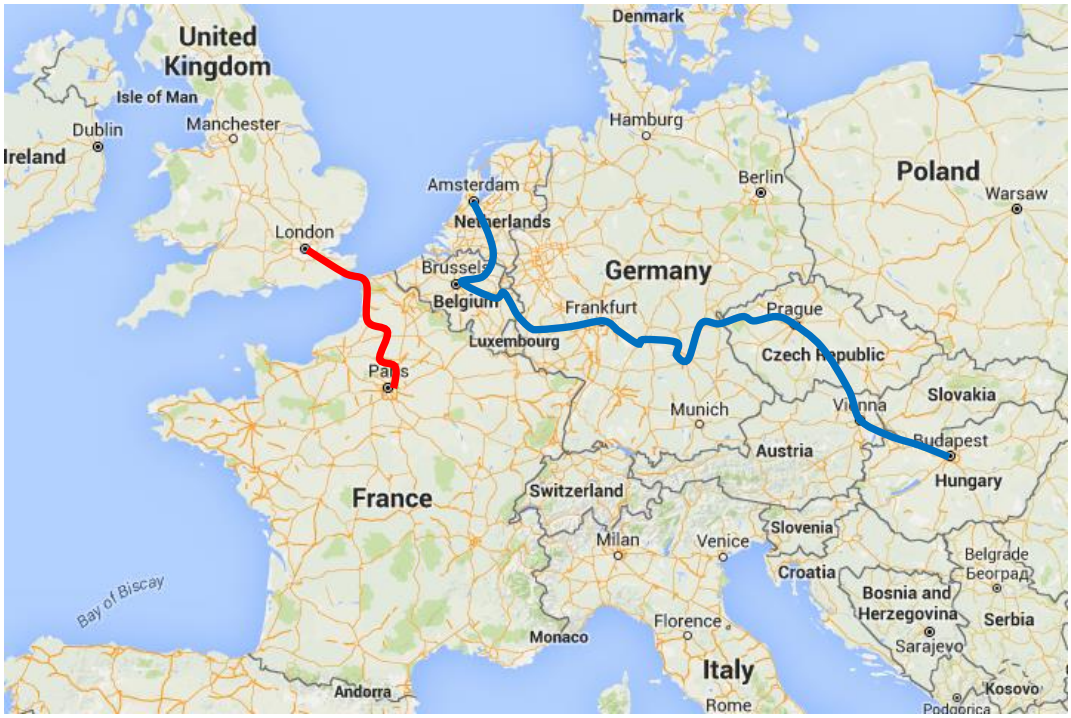

Figure1. Europe Map (Google maps)

Figure1 shows a map of Europe. The red line path in Figure1 is a sample tour path from London to Paris. The path only has two cities and there is no stopover city. The blue line path is a second sample path containing stopover cities such as Brussels, Frankfurt, Prague, and Vienna. We want to find a shortest tour path to minimize the total moving time. (Each path between cities has a fixed time to pass.)

You should suggest an efficient algorithm to solve PA2.

Input

Your program is to read from standard input. The first line of the input gives the number of descriptions about travel time, **E**. **E** lines follow. Each line consists of names of two cities and the time required to travel from one city to the other. **(Our input is an undirected graph)**

The next line gives the number of test cases, **T**. **T** test cases follow. Each test case begins with a line containing 3 arguments: number of stopover cities **s**, departure city, and arrival city. Then the s lines follow and contain the name of stopover cities. **(Each city is a string whose length is less than 20, the time is a positive integer, and the number of stopover cities is less than 20)**

Output

Your program is to write to standard output. For each test case, output consists of 3 parts: test case number (starting from 1), the least total travel time, and travel path(s). If there are more than one path, describe all paths. **Wrtie "no path" if there is no path between cities including stopover cities**.

The following shows sample input and output for two test cases.

Sample Input	Output for the Sample Input
16 Berlin Amsterdam 4 Berlin Frankfurt 1 Berlin Praha 2 Berlin Zurich 9 Amsterdam Frankfurt 2 Amsterdam London 4 Amsterdam Paris 5 Frankfurt Zurich 7 Praha Paris 11 Praha Wien 6 London Paris 3 London Rome 6 Paris Zurich 1 Zurich Wien 5 Zurich Rome 4 Wien Rome 6 2 0 Berlin Rome 2 Wien London Berlin Zurich	case 1 12 Berlin Frankfurt Zurich Rome case 2 20 Wien Praha Berlin Frankfurt Zurich Paris London Wien Zurich Frankfurt Berlin Frankfurt Amsterdam London