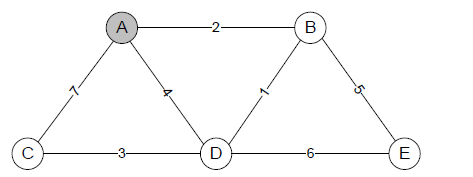
**First Name:**  Artur  **Last Name:** Gordiyenko

* Please type you answers on this worksheet with blue color and submit it to Lab-9 on D2L prior to the next week class.
* Please do not zip or compress your submissions. D2L allows you to upload multiple files.

**Queston1:**

[2 mark] Apply Dijkstra's algorithm to the following graph using vertex A as the source. Show the distance vector and parent array after each iteration of the algorithm.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Vertex | A | B | C | D | E |
| ***Parent*** |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Parent*** |  |  |  |  |  |

*initialization 0*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Vertex | A | B | C | D | E |
| ***Distance*** |  |  |  |  |  |

*iteration 1 2 7 4 A*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Distance*** |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Parent*** |  |  |  |  |  |

*iteration 2 7 3 7 A B*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Distance*** |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Parent*** |  |  |  |  |  |

*iteration 3 6 7 A B D*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Distance*** |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Parent*** |  |  |  |  |  |

*iteration 4 7 A B D C*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Distance*** |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Parent*** |  |  |  |  |  |

*iteration 5 A B D C E*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Distance*** |  |  |  |  |  |

**Queston2:**

**Golden State Motorcycle Riders**

The Golden State Motorcycle Riders Association (GSMRA) has decided to provide a trip routing service to its members. They have asked you to write a program which reads a list of departure point-destination point pairs and calculates the shortest routes between them. For each trip, your program will print a report which itemizes the names of each city passed through, with route names and leg distances.

**Input**

Input to your program will be in two parts.

The first part is a map in the form of a list of highway segments. Each segment is designated by a line containing four fields which are separated by commas. The first two fields are 1-20 characters each, and are the names of the cities which are at each end of the highway segment. The third field is the 1-10 character name of the route. The fourth field is the number of miles between the two endpoints, expressed as a positive integer. The highway segment list will be terminated by an empty line.

The second part of the input is a list of departure point-destination point pairs, one per line. The departure point is given first, followed by a comma and the destination point. Each of the cities is guaranteed to have appeared in the first part of the input data, and there will be a path that connects them. The list is terminated by the end of file.

**Output**

The output should be a series of reports, one for each departure point-destination point pair in the input. Each report should be in exactly the same form as those in the example below. There should be two blank lines before each report (including the first one).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sample Input**  San Luis Obispo,Bakersfield,CA-58,117 | **Sample Output**  From | To | Route | Miles |
| Bakersfield,Mojave,CA-58,65 | -------------- | ------------- | ------- | ----- |
| Mojave,Barstow,CA-58,70 | Santa Barbara | Los Angeles | US-101 | 95 |
| Barstow,Baker,I-15,62 | Los Angeles | San Bernardino | I-10 | 65 |
| Baker,Las Vegas,I-15,92 | San Bernardino | Barstow | I-15 | 73 |
| San Luis Obispo,Santa Barbara,US-101,106 | Barstow | Baker | I-15 | 62 |
| San Luis Obispo,Santa Barbara,CA-1,113 | Baker | Las Vegas | I-15 | 92 |
| Santa Barbara,Los Angeles,US-101,95 |  |  |  | --- |
| Bakersfield,Wheeler Ridge,CA-99,24 |  |  | Total | 387 |
| Wheeler Ridge,Los Angeles,I-5,88 |  |  |  |  |
| Mojave,Los Angeles,CA-14,94 |  |  |  |  |
| Los Angeles,San Bernardino,I-10,65 | From | To | Route | Miles |
| San Bernardino,Barstow,I-15,73 | -------------- | ------------- | ------- | ----- |
| Los Angeles,San Diego,I-5,121 | San Diego | Los Angeles | I-5 | 121 |
| San Bernardino,San Diego,I-15,103 |  |  |  | --- |
|  |  |  | Total | 121 |
| Santa Barbara,Las Vegas |  |  |  |  |
| San Diego,Los Angeles |  |  |  |  |
| San Luis Obispo,Los Angeles | From | To | Route | Miles |
|  | -------------- | ------------- | ------- | ----- |
|  | San Luis Obispo | Santa Barbara | US-101 | 106 |
|  | Santa Barbara | Los Angeles | US-101 | 95 |
|  |  |  |  | --- |

Total 20

Answer the following questions for the above problem.

1. [1 mark] What type of problem is this (i.e.: shortest path, topological sort, spanning tree, union find, etc.)?

Shortest path

1. [1 mark] Which algorithm (that you know) can be used (with enhancements) to solve this problem? (if more than one algorithm can be used, explain why you selected the one that you did)

Dijkstra's algorithm, because we need to value the distance between each Vertex. So Depth First Search would not work properly in this case.

1. [2 mark] Explain how you can model the problem as a graph. What do vertices, edges etc. represent? Draw the graph that you corresponds to the sample input that is provided with the problem.

Each Vertex will represent a city.

Each Edge will represent a highway.

Vertices will contain lists of Edges (or highways) connected to them.

Edges will contain the two Vertices (or cities) that they connect.

Graph Attached as "question2graph.png"

**Question3**

**Solozol**

The independent nation of Solozol is perfectly flat. Unfortunately, Solozol has a very poor system of public highways. The Solozolian government is aware of this problem and has already constructed a number of highways connecting some of the most important towns. However, there are still some towns that you can't reach via a highway. It is necessary to build more highways so that it will be possible to drive between any pair of towns without leaving the highway system.

Solozolian towns are numbered from 1 to N and town i has a position given by the Cartesian coordinates (xi, yi). Each highway connects exactly two towns. All highways (both the original ones and the ones that are to be built) follow straight lines, and thus their length is equal to Cartesian distance between towns. All highways can be used in both directions.

Highways can freely cross each other, but a driver can only switch between highways at a town that is located at the end of both highways. The Solozolian government wants to minimize the cost of building new highways. However, they want to guarantee that every town is highway-reachable from every other town. Since Solozol is so flat, the cost of a highway is always proportional to its length. Thus, the least expensive highway system will be the one that minimizes the total highways length.

**Input**

The first line of the input consists of an integer indicating the number of test cases in the input. Then there's a blank line and the datasets separated by a blank line.

Each test case consists of two parts. The first part describes all towns in the country, and the second part describes all of the highways that have already been built.

The first line of the test case contains a single integer N (1 ≤ N ≤ 750), representing the number of towns. The next N lines each contain two integers, x i and y i separated by a space. These values give the coordinates of i Th town (for i from 1 to N). Coordinates will have an absolute value no greater than 10000. Every town has a unique location.

The next line contains a single integer M (0 ≤ M ≤ 1000), representing the number of existing highways. The next M lines each contain a pair of integers separated by a space. These two integers give a pair of town numbers which are already connected by a highway. Each pair of towns is connected by at most one highway.

**Output**

Write to the output file a single line for each new highway that should be built in order to connect all towns with minimal possible total length of new highways. Each highway should be presented by printing town numbers that this highway connects, separated by a space.

If no new highways need to be built (all towns are already connected), then the output file should contain a line with the sentence "No new highways needed". Print a blank line between test cases.

**Sample input**

1

9

1 5

0 0

3 2

4 5

5 1

0 4

5 2

1 2

5 3

3

1 3

9 7

1 2

**Sample output for the sample input**

1 6

3 7

4 9

5 7

8

Answer the following questions for the Solozol problem.

1. [1 mark] What type of problem is this (i.e.: shortest path, topological sort, spanning tree, union find, etc.)?

Minimum spanning tree

1. [1 mark] Which algorithm (that you know) can be used (with enhancements) to solve this problem? (if more than one algorithm can be used, explain why you selected the one that you did)

Prim algorithm

Kruskals algorithm.

I chose prim algorithm because its efficiency is better than Kruskals O(|E|) vs O(|E|log|E|)

1. [2 mark] Explain how you can model the problem as a graph. What do vertices, edges etc. represent? Draw the graph that you corresponds to the sample input that is provided with the problem.

Just like in previous problem, the edges will represent highways connecting two vertices (cities)

Vertices will represent individual cities.

Graph Attached as "question3graph.png"