**Name: Dongsun Kim**

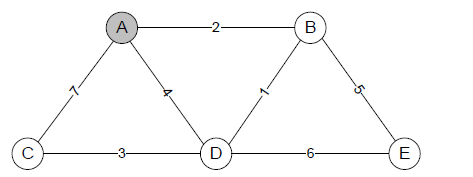
**ID: A01052311**

**Set: 3O**

Please write your answers in this Word document and submit it to Learning Hub by the deadline (as noted in Learning Hub).

**Question 1:**

[5 points] Apply Dijkstra's algorithm to the graph below using vertex A as the source. Show the distance vector and parent map after each iteration of the algorithm. ∞



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

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

*initialization* 0

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

*iteration 1*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Distance*** | 0 | 2 | 7 | 4 | ∞ |
|  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Parent*** |  | A | A | B | B |  |

*iteration 2*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Distance*** | 0 | 2 | 7 | 3 | 7 |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Parent*** |  | A | D | B | B |

*iteration 3*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Distance*** | 0 | 2 | 6 | 3 | 7 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Parent*** |  | A | D | B | B |

*iteration 4*

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

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

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

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Distance*** | 0 | 2 | 6 | 3 | 7 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Parent*** |  | A | D | B | B |

*iteration 5* 0 2 6 3 7

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

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

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

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

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

**Question 2:**

**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 201

Answer the following questions for the above problem:

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

Shortest path

1. [1 point] 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

1. [1 point] Explain how you can model the problem as a graph. What do vertices, edges etc. represent?

Vertices = City names

Edges = Routes

Miles = value of edges

1. [2 points] Draw the graph that corresponds to the sample input that is provided with the problem.

Note: You may draw this graph by computer (any means you have available) or by hand (on paper). But whatever method you use to draw the graph, **please create an image file from the result and paste the image into this Word document**.

A close up of a map

Description automatically generated

1. [**Optional**] Implement the Java code for the problem.