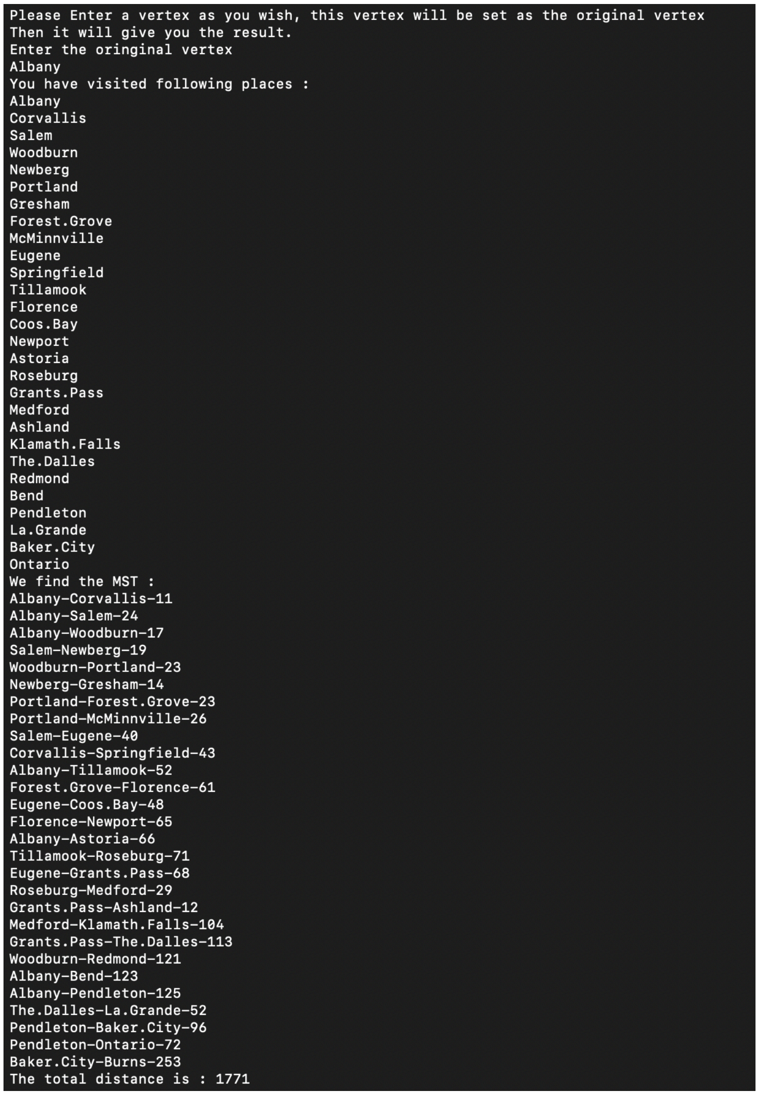
CS350
HW4
Hanyang Xiao-scetion 006
Jiancheng Zhao-section 004

To implement Prim’s algorithm, we choose C++ since it is my most familiar language. Another reason is that we implement some algorithms when we took CS163 so it is easier for me this time. [Build Instruction] - Adjacent list as the data structure since we believe it is easier to represent a weighted graph. Each vertex is stored as an element in the adjacency list also with a head pointer which points at a linear linked list. Thus, edges and weights will be stored in the linear linked list. The source we use is the textbook and we found some details there.
For the second question, it is harder than the first one to implement it. The run instructions is already put at the beginning of our program and you will find it as soon as you run it. The picture is example of our output below.



```
Please Enter a vertex as you wish, this vertex will be set as the original vertex
Then it will give you the result.
Enter the oringinal vertex
Albany
You have visited following places :
Albany
Corvallis
Salem
Woodburn
Newberg
Portland
Gresham
Forest.Grove
McMinnville
Eugene
Springfield
Tillamook
Florence
Coos.Bay
Newport
Astoria
Roseburg
Grants.Pass
Medford
Ashland
Klamath.Falls
The.Dalles
Redmond
Bend
Pendleton
La.Grande
Baker.City
Ontario
We find the MST :
Albany-Corvallis-11
Albany-Salem-24
Albany-Woodburn-17
Salem-Newberg-19
Woodburn-Portland-23
Newberg-Gresham-14
Portland-Forest.Grove-23
Portland-McMinnville-26
Salem-Eugene-40
Corvallis-Springfield-43
Albany-Tillamook-52
Forest.Grove-Florence-61
Eugene-Coos.Bay-48
Florence-Newport-65
Albany-Astoria-66
Tillamook-Roseburg-71
Eugene-Grants.Pass-68
Roseburg-Medford-29
Grants.Pass-Ashland-12
Medford-Klamath.Falls-104
Grants.Pass-The.Dalles-113
Woodburn-Redmond-121
Albany-Bend-123
Albany-Pendleton-125
The.Dalles-La.Grande-52
Pendleton-Baker.City-96
Pendleton-Ontario-72
Baker.City-Burns-253
The total distance is : 1771
```

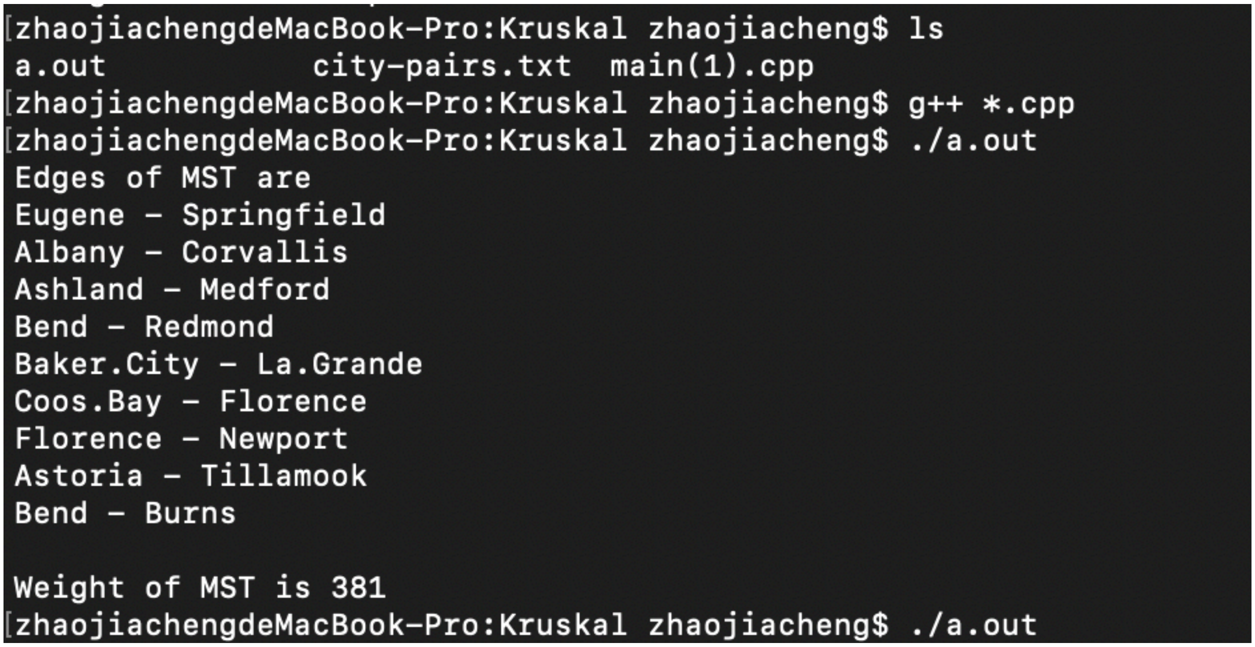
Extra credit (Kruskal's algorithm)

To implement Kruskal's algorithm, we also choose C++ with same reason. Created weighted and undirected graph with vector which we would like use different data structure with Prim's algorithms. Each vertex store an element. The edge stored the relation with each two elements. Created a structure to make sure all element is connected as result. The source we use is the textbook and we found some details there.

It is hard to learn create graph by vertex, and this one is harder than Prim's algorithm when dealing with save visited graph.

The run instructions is already put at the beginning of our program and you will find it as soon as you run it.

The picture is example of our output below.



```
[zhaojiachengdeMacBook-Pro:Kruskal zhaojiacheng$ ls
 a.out           city-pairs.txt  main(1).cpp
[zhaojiachengdeMacBook-Pro:Kruskal zhaojiacheng$ g++ *.cpp
[zhaojiachengdeMacBook-Pro:Kruskal zhaojiacheng$ ./a.out
 Edges of MST are
 Eugene - Springfield
 Albany - Corvallis
 Ashland - Medford
 Bend - Redmond
 Baker.City - La.Grande
 Coos.Bay - Florence
 Florence - Newport
 Astoria - Tillamook
 Bend - Burns

 Weight of MST is 381
[zhaojiachengdeMacBook-Pro:Kruskal zhaojiacheng$ ./a.out
```