PROJECT #4: Dijkstra’s Algorithm

PRELIMINARY ANALYSIS AND DESIGN

Problem Description

Building a directed graph of vertices from the data stored in an input file, print out a listing of vertices contained in the graph in sorted order (three per line), produce a summary table of the shortest paths from the starting vertex input by the user to all other vertices.

Specification

* Using C++ coding
* Using the graph.h specification file to implement Graphing tools
* Using the SeparateChaining.h specification file to implement Hashing
* Radix Sort to sort list of vertices in alphabetical order

Steps to store the data from the input file:

* Step 1 : Store the first city name, second city name and weight in the variables
* Step 2: A while loop where the city1, city2 are added as vertices to a Graph
* And the while loop is repeated with the next two city names and their weight is inputted step is repeated until there are no more inputs;
* Step 4: The Vertices are sorted using the Radix Sort
* Step 5: The Vertices are outputted in alphabetical order.

Dijkstra’s Algorithm

* Determine the vertices which are adjacent to your current vertex
* Determine the weight of the adjacent vertices and the current vertex
* If andjacent vertex is unmarked and its distance value in the table is greater than the sum of the weight value plus the distance of the current vertex
  + Reset the distance of the adjacent vertice to the smaller sum
  + Store the name of the current Vertex as the previous Vertice of the adjacent vertices
* Find the minimum distance of all unmarked vertices, and set the vertex that corresponds to the minimum distance as your new current vertice
* Print out the data corresponding to your current marked vertex
* Repeat the previous 4 steps until all of your vertices have been marked.

Rambaugh Diagram