Matt Facque

Albert Galang

CS 302 – Data Structures

Design Document

Classes:

Class Weighted Graph

- Adjacency list : vector<node>

+ Add(vertex, edges) : bool

+ Brute Force Traversal (City, void visit(&T)) : vector<path, total mileage>

Class Node

- city : string

- miles : int

- next node : node\*

+ GetCity() : string

+ GetMiles() : int

+ SetCity() : void

+ SetMiles(): void

+ GetNextNode() : node\*

Main () {

Build City graph

Initialize graph

Add “RENO”…..

Possible\_paths = graph::Brute\_Force\_Traversal(starting City);

Int lowest\_miles = 0;

For () {

If (possible\_paths[i] < possible\_paths[lowest\_miles]) {

Lowest\_miles = I;

}

}

Display(possible\_paths);

Std::cout << “Lowest cost: “ << std::endl;

Display(possible\_paths[lowest\_miles]);

Return 0;

}

This is basic version of our driver, initialize the graph. Assign cities and thereby create weighted edges. The Brute\_Force\_Traversal() function is what determines the routes and the total mileage of the routes. We then compare those vectors in a for loops and then output all routes and mileage, then the best route.

Responsibilities

Albert -> Weighted Graph Class

Matt -> Node Class and driver

Timeline

1. Weighted Graph and Node class completed on Friday, May 1st
2. Driver implemented on Saturday, May 2nd
3. Debugging Sunday through Monday
4. Testing and Refinement Tuesday, May 5th
5. Complete and turn in, Wednesday May 6th

Albert and I have already had numerous phone conversations as well as screen sharing over discord and we will be putting out code together on github. So far we haven’t had any problems, although the school semester is winding down and getting busier, we both seem to be working together quite well.