Project Specification:ENEE641

Due: November 29

Objective:

Given an undirected connected graph G = (V,E) where each edge has a weight, find a minimum spanning tree. Suppose that vertices are cities, the weight of each edge is the cost of connecting two cities and a minimum spanning tree (MST) would provide the minimum cost of connecting together all cities. You need to write 2 programs, one implementing Prims algorithm and the other implementing Kruskals algorithm.

Input Format:

In the benchmarks, cities are numbered starting from 1. The x and y co-ordinates of all the cities are given in the format "city-number x-coordinate y-coordinate".A list of pairs of cities is provided and the edge weight between each such pair of cities is simply the Euclidean distance.

Output:

The following is expected from you for each algorithm:

1. Your algorithm and its explanation
2. Asymptotic analysis of the runtime of your algorithm up to a constant factor, counting C command (see below how to report this in your program)
3. Experimental results for the benchmarks provided. Report the cost of the MST as well as actual runtime of the program in terms of C command executed (see below)

Deliverables:

- Source code compatible on glue, with the output files for the benchmarks
- At most 4 page report illustrating point [1,2,3] outlined above.

Output format:

A:

Your program should produce an output file containing the following output format:

city-number1 city-number city-number .....
city-number2 city-number city-number .....
.
.
city-numberN city-number city-number .....

The output format can be understood in the following manner. Each line will contain the connectivity information about a particular city. Like in this example:

1 2 3
2 1 4
3 1
4 2

City number 1 is connected to 2 and 3, and 2 is connected to 4. So the total cost of connection will be the cost of edge between 1 and 2, plus the cost of edge between 1 & 3 and the cost of edge betwen 2 & 4.

B:

Your program should produce an output file containing the following output format:

City (vertex) number 1
City (vertex) number 2
.
City (vertex) number n
Pair of cities (edge) number 1
Pair of cities (edge) number 2
.
Pair of cities (edge) number m

For each of the n+m lines print out in unary the total number of operations that have been charged to that item. The printed out numbers should match as closely as possible with your asymptotic analysis.

Benchmarks:

Benchmarks will be available on the class website by November 4