CSc130 Assignment 4
Prim’s MST Algorithm

Implementation of Prim’s MST Algorithm using Priority Queue:

The purpose of this assignment is to give you a practice in programming graph algorithms and priority queue.

You are to write a robust, well-structured and well-documented program to implement Prim’s algorithm using Java. “Robust” means that error-checking is done on the inputs and the program does not crash. “Well-structured” means that you should follow the principles of good design: modular set of classes and functions, no function should be too long, and each function/class should correspond to a well-defined and cohesive task/concept. “Well-documented” means that you need to give overview comments for each function and class, and you need to explain all aspects of the code that are not obvious.

- The input graph is to be read in from a file. The format of the file is this: The number of vertices, n, is given on the first line of the file. The vertices are {1, 2, … , n}. Each subsequent line contains two integers, each between 1 and n, indicating the existence of an edge between these two vertices, followed by a real number, indicating the weight on the edge. The graph is undirected. For example. A triangle graph in which all the weights are real numbers would be represented like this:

  ```
  3
  1    2  10.5
  2    3  11.3
  3    1  100.2
  ```

- To test your program, you may use the graph on text book page 604, tinyEWG.txt. That input file contains the number of edges and our input file does not. Note that grader will test your code using different graph too; therefore, your code should work correctly for any connected undirected graph.
- The output for the algorithm is the list of edges (and their weights) in a MST computed by the Prim’s algorithm as well as the total weight of the MST.
- Textbook website has sample implementation Java codes for both PrimMST and KrusalMST. We also prepared a reference for this assignment which contains a comparative analysis of both MST algorithms and a sample implementation Java code for Krusal’s algorithm.
- In your implementation of Prim’s algorithm, implement the priority queue with a binary heap.

Deliverables:

Design file: Well written algorithm or flow chart of your program.

Source code: well commented source code in both pdf and Java file separately. Your pdf file should contain source code and easy to read. Your Java file should be ready to run for grader to test with new input data in required format.

Sample input and output