CS3310: Programming Assignment VI
Minimum Spanning Trees and Networks
(Due time: 11:59:59pm 4/16)

Concepts

- Constructing Minimum Spanning Trees
- Disjoint Sets (union-find structures)
- Kruskal's Algorithm

Background

A graph is a data structure that contains nodes called vertices and links between vertices, called edges. Edges can be weighted and have possible directionality, informing how an edge should be traversed in a graph. A type of graph, called the minimum cost spanning tree, is where all vertices are connected by minimum cost undirected edges, to form a tree. Since the structure is a tree, there are no cycles between connected vertices. There are many applications for minimum cost spanning trees (MSTs), including circuit diagram representation construction and communication network design.

Problem Specification

In this assignment, you will construct a minimum cost spanning tree using Kruskal's algorithm that represents a network for providing communications access between cities on a graph. To ensure no cycle is formed in Kruskal's algorithm, a disjoint set structure is required. A disjoint set relates to two methods, i.e., union() and find(). Union() requires the indices of two tree roots in a disjoint set and possibly the disjoint set, then creates a larger tree by setting one root as the parent of another. In the find() method, a vertex index in the set is passed as an argument and possibly the disjoint set, then the index of the root of its tree must be found and returned. Due to being provided extensive source code from linked web sources on eLearning for Weeks 12 and 13 lecture materials, you can reference and use sources in your project (i.e. GeeksforGeeks, programiz, etc.). You must comment and acknowledge where the referenced code originated, to uphold academic honesty. Since this assignment will gauge your understanding about how minimum spanning trees and Kruskal's algorithm work, you must do the following tasks:

1. Hard code a graph structure that represents a grouping of generic cities on a map, with several vertices and weighted edge instances
2. Implement Kruskal's algorithm to build a minimum cost spanning tree (should have no cycles) from a graph, which represents the optimal communications network configuration.

3. Print the resulting built MST edges
4. Determine which city (node) is best suited to be the central hub of the network by picking the city (node) whose largest path cost to all other nodes is minimum, compared to any other node being chosen as a potential hub. Print which is the best city (node) to choose and what is the largest path cost out to the furthest node.

Note here, once the minimum cost spanning tree is found, an optimal least-cost communication network is created between cities. To determine the best access point to connect to all other cites, you must find the city node on minimum cost spanning tree whose largest path cost to the other nodes is minimum, in comparison to choosing all other nodes as hubs. In case multiple cities have tying minimal largest path costs out to the furthest node, choose one as the central hub.

You must follow the software engineering practices to write your project. The practices include:

1. (Package name, class name for Java), (Module name, class name for Python), and
2. Encapsulated attributes in a class
3. Well commented/documented code.

Implementation Phase

Now you will create the actual program. You must work on the rest of the assignment *individually*.

Testing Phase

Java main method

```
public class Main {

    public static void main(String[] args) {

        //Initialize a graph with hard coded vertices and edges of your choice

        //call Kruskal's implementation to complete the min cost spanning tree and return
        // minimum cost.
        // find which node in the optimal network should be the central hub
        // print which node should be chosen as the central hub and what is the largest cost out
        // to the furthest node



    }

}
```

Python main.py

```
If __name__ == "main":

    # Initialize a graph with hard coded vertices and edges of your choice

    # call Kruskal's implementation to complete the min cost spanning tree and return
    # minimum cost.
    # find which node in the optimal network should be the central hub
    # print which node should be chosen as the central hub and what is the largest cost out to the
    # furthest node
```

Assignment Submission

- Follow the website (http://www.cs.wmich.edu/%7Ewwshen/cs1120_Fall2012/documents/HowtoSubmitPAinEclipse.pdf) to export your project in a .zip file that contains all of your files, including:
    - Program Files

    - Including all source programs
    - including any input or output files
- Submit the .zip file via E-learning (A dropbox will be created).