Computer Science 2300: Lab 4

Due: February 20, 2008

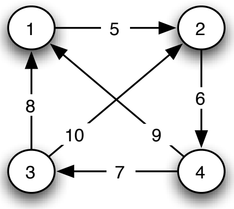
For this lab you will be implementing Kruskal's algorithm. Make sure you read all instructions carefully.

1 Implementing Kruskal's algorithm

Write a program that implements Kruskal's algorithm **using a disjoint set data structure**. The program takes as input a file (`data.txt`), in which each line either represents a vertex or an edge. For the edge lines, the first integer on that line representing the starting vertex, the second the ending vertex, and the third the weight of the edge. Use this file to construct, line by line, the graph upon which Kruskal's algorithm will be run (do NOT hardcode this graph!). For example, consider the following text file:

```
<sample.txt>
V 1
V 2
V 3
V 4
E 1 2 5
E 2 4 6
E 4 3 7
E 3 1 8
E 4 1 9
E 3 2 10
```

The graph corresponding to this text file is the following:



For your convenience, a graph class has been provided for you (`lab6-files.tar.gz`), as well as a rough implementation of the disjoint set class. You have two tasks:

1. First, implement the `find()` and `unifyWith()` functions in the file `disjointSet.h`.
2. Implement Kruskal's algorithm using your complete disjoint set class.

Edit the file `main.cpp` found in this tar-file to implement Kruskal's algorithm. You may additionally modify the graph class in whatever way you wish in order to complete this lab (with the exception of hardcoding solutions). **Note: you may have to implement additional code to get your solution to fully work**.

2 Compiling and Running Your Code

In addition to `data.txt`, a basic Makefile is included in `lab6-files.tar.gz`. To compile your program, simply type in the command

```
make
```

The makefile will then compile your code with the graph class, and create an executable, `kruskal`. For your convenience, `main.cpp` originally contains some starting functions to help you get familiarized with the provided graph class and disjoint set data structure.

To receive full credit, run Kruskal's algorithm on the graph generated by file `data.txt`. Show your output to the TA to receive full credit. You can re-use any code you like from lab 4; the (`disjointSet.h` class contains the class and its definition, and thus is standalone).