BLG 372E - Analysis of Algorithms, Spring 2013

Project 2

(Please read report and submission part carefully)

Total Worth : 10% of your grade

Handed Out : 04.04.2013 16:00

Due : 18.04.2013 23:00

Overview

We are given a collection of cities. There is also a collection of directed links between cities, with flight costs along the links. Kruskal's algorithm will be used to find a minimum spanning tree for a given graph which is based on the city and directed link collections. Three different tests, from a sparse graph to a dense graph, will be run and results will be analyzed.

Algorithm

You should implement **Kruskal's minimum spanning tree algorithm** with **disjoint set** data structure. You should use C++ language. In the first test, you should choose *k* random links and build a graph with chosen links. Then you should find MST for this graph. In the second test, you should **add** *k* more randomly chosen links to the first graph and find MST again. In the last test, you should **add** *k* more links and find MST again. So, the algorithm should be run on graphs with *k*, *2k* and *3k* links.(You need to examine if choosen k links form a connected graph or not, a connected graph should be formed. Do not add the complexity of this calculation to the total algorithm complexity.)

Algorithm should keep a count of "basic steps". What your basic step is will depend on you, but it should be something that is proportional to the total work the algorithm does, such as number of edges visited. Algorithm should also keep the running time of each case. If the running time is too short, you can find the MST for a graph multiple times, for example 100 times and report the total running time divided by 100.

For each test your algorithm should output the initial graph (as a collection of edges (u,v) where u and v are node indices), MST it finds, along with the distance obtained with it, the running time, and the number of basic steps required. You should summarize your results in two plots, showing the number of edges on the x axis and the running time or the basic steps required on the y axis.

Data and Test Runs

There are three datasets in three distinct folder named as data10, data40, data80. A dataset will consist of two files, one with a list of cities (Cities_[10,40,80].txt), and another with links and their costs (Flights_[10,40,80].txt). Both files are comma separated text files.
Cities_[10,40,80] format : city_name.
Flights_[10,40,80] format: source, destination, cost.

Report, Submission

You should be aware that the Ninova system clock may not be synchronized with your computer, watch, or cell phone. Do not e-mail the teaching assistant or the instructors your submission after the Ninova site submission has closed. If you have submitted to Ninova once and want to make any changes to your report, you should do it before the Ninova submission system closes. Your changes will not be accepted by e-mail. Connectivity problems to the Internet or to Ninova in the last few minutes are not valid excuses for being unable to submit. You should not risk leaving your submission to the last few minutes. After uploading to Ninova, check to make sure that your project appears there.

Policy: You may discuss the problem addressed by the project at an abstract level with your classmates, but you should not share or copy code from your classmates or from the Internet. You should submit your own, individual project. Plagiarism and any other forms of cheating will have serious consequences, including failing the course.

If a question is not clear, please let the teaching assistant know by email (nilhan@itu.edu.tr).

Submission Instructions: Please submit your homework through Ninova.

Please zip and upload all your files using filename **HW2_ studentID.[zip,rar]**. In the zipped file, you must include your completed **report_StudentId** file, a folder named as "**Codes**. Test your programs with the files given to you.

Your compilation code will be:

```
g++ main .cpp –o main
```

Example run for task1:

```
./main "Cities_10.txt"  "Flights_10.txt"  "result10.txt"
```

The report should clearly explain the Kruskal's algorithm, complexity of your project, test results and analysis of results according to the complexity. Give the data structures you have built and explain them. Give the compilation command of your program. Be sure that your program can be run with the running command given above. Otherwise, you may get a grade of zero.

All your code must be written in **C++,** and we should be able to compile and run on it on Linux/Unix using **g++**. When you write your code, try to follow an object-oriented methodology with well-chosen variable, method, and class names and comments where necessary. Your code must compile without any errors; otherwise, you may get a grade of zero on the assignment.