

CS 302, Spring 2007
Assignment #6
His and Hers Graph Algorithms
to be done in pairs!

Due 4/24/07, in class

Introduction

Graphs can be represented using an adjacency matrix or adjacency list. This project investigates these two as an example of a classic time/space trade-off. Write two programs: `graph-gen` and `graph-stats`. The program `graph-gen` should output the vertices and then the edges of a graph. For example, K_3 would be output as (The first line in the vertex count. The “-1” lines are sentinels.)

```
3
0 N0
1 N1
2 N2
-1 *
0 1
0 2
1 2
-1 -1
```

You can assume that the vertex numbers run from 0 to the *number-of-vertices-1* and that the vertex names are no more than 255 characters and contain no white space.

Assignment

Implement the Euclidean Neighbor algorithm in `graph-gen` and then use it to generate multiple graphs having 1000 and 10,000 vertices. For each size use the distances 0.1, 0.3, 0.5, 0.7, and 0.9.

Then implement `graph-stats`, which reads in a graph from stdin and outputs various stats about the graph. It should take as a command line argument the switch `-list` if an adjacency list representation is to be used and `-bitset` (the default) if the adjacency matrix representation is to be used.

For each of the ten graphs generated by `graph-gen` output the following:

- 1) number of vertices
- 2) number of edges
- 3) vertex with the most incident edges
- 4) the time taken to initialize the graph (pre reading)
- 5) the time taken to read the graph
- 6) the time taken to process the graph
- 7) the memory foot-print of your program right before termination.

Finally, create a 1-2 page well laid out summary report of the data (include some nice looking graphs).

What to hand in

- (1) (by 4/13) A one-page design.
- (2) (by 4/13) A 2up print out of your black-box test plan.
- (3) (on 4/27) A 2up well-formatted listing of your code.
- (4) (by 4/27) Email me a single tar file that includes
 - the file `Inventory`
 - all your source files
 - your `Makefile`
 - any auxiliary files that you have accumulated **DO NOT** include the generated random graphs!
If you did not use a pipe, then delete them when you are done as some will be very large.
 - the output from executing your program on this test plan (number each test).
 - your 1-2 page summary