

Graph Mining Lab -3 - Readme

Instructions:

Q1.

In the given question, we need to generate an orientation based upon (1) degree of the vertices (2) graph degeneracy (3) number of triangles contained per vertex. For breaking a tie, use the vertex id and Report the maximum indegree and maximum outdegree as the output of your implementation for graph-1 and graph-2.

NOTE:

Make sure that input files (graph-1.txt and graph-2.txt) are in the same directory as the code of execution. The code is taking graph-1 as input by default. To run graph-2 please change the file that is being read to "graph-2.txt"

```
g++ Q1.cpp  
./a.out
```

Q2.

In the given question, we need to implement an algorithm for counting all cliques of size k in the input graph from the paper "Arboricity and Subgraph Counting" by Chiba and Nishizeki and implement an algorithm for counting all maximal cliques from the same paper.

NOTE:

This code is written in C language. The graph which is to be given as input should be renamed as **edgelist.txt**. Both the input k and the graph are to be given in the command line argument. Run the code on a Linux machine with a GCC Compiler.

```
gcc q2.c -O9 -o q2
```

`./q2 k edgelist.txt`

Q3.

In the given question, we need to implement the Community Detection algorithm due to Girvan & Newman discussed in the class.

NOTE: Make sure that input files are in the same directory as the code of execution.

Run the code on a Linux machine with a GCC Compiler.

Command Line 1 : `g++ Q3.cpp`

Command Line2 : `./a.out`

Q4.

In the given question, we need to implement the implement Node classification using the label propagation technique discussed in the class.

NOTE: Make sure that input files are in the same directory as the code of execution. The code is taking graph-3 as input by default.

Run the code on a Linux machine with a GCC Compiler.

Command Line 1 : `g++ Q4.cpp`

Command Line2 : `./a.out`