Assignment 3 -- Programming Project

Design and Analysis of Algorithms (CS 4071) -- Fall 2016

Diameter and Connected Components of a Network

Electronic copy of source code due Sunday, Nov. 6 at 12:00 midnight. Hard copy due at beginning of class on Monday, Nov. 7. Include your group number and members of the group as a comment at the beginning of the program and submit only ONE hard copy and ONE electronic copy (have ONE group member upload electronic copy).

Topics covered: *Graphs, BFS, distance matrix, diameter, connected components.*

The topology of a network is modeled with a graph. Write a C++ program that inputs a graph G by first inputting the number of vertices n followed by a sequence of pairs i j where i and j are integers between 0 and n, inclusive, representing the edges of the graph, and ending with a negative integer sentinel to indicate the end of the input. For example,

```
5  0  1  1  4  2  3  1  3  3  4  -1
```

represents the graph G = (V,E) given by:

V = {0,1,2,3,4}

E = {{0,1},{1,4},{2,3},{1,3},{3,4}}.

Your program will compute the **distance matrix** of G, which you will use to compute the **diameter** of G in the case when G is connected and the **connected components** of G, otherwise.

You can proceed as follows:

- Implement the graph G with its **adjacency matrix**
- Implement the function `BFS(G,v)` for performing a **breadth-first search** where the visit operation involves computing the distance from v to the vertex being visited. This will require a queue, which you can get from the Standard Template Library (STL).
- Implement a function `DistanceMatrix(G)`, which computes the **distance matrix** of G, where entry ij is assigned the distance from i to j if i and j are connected and -1 otherwise. Have `DistanceMatrix()` also returns a Boolean value that is true if G is connected and false, otherwise. `DistanceMatrix()` will call `BFS()`.
- Implement a function `Diameter(G)` that returns the **diameter** of G if G is connected and -1, otherwise. `Diameter(G)` will call `DistanceMatrix()`.

- Implement a function `Components(G)` for computing the vertex sets of the connected components of G. `Components()` will call `DistanceMatrix()`.

Store the entire source code for your program in a **single** file. Your program should run using g++ on a UNIX platform.

Have your program output the adjacency matrix of the graph as well as the distance matrix, the diameter in the case when G is connected and the connected components, otherwise. Output the connected components by outputting the vertex set of each connected component.

Your program should be **user-friendly**, **well-commented**, with the output **well-documented**.

Submit the source code for your program on-line by uploading to Blackboard (have ONE member of your group upload. Make sure to include a comment in your program with your Group Number and Group Members). Also submit a **hard copy** of your program with output for **two** sample runs, one with a connected graph G and one with a disconnected graph G (ONE hard copy per group submitted at the beginning of class on the due date).