CSCI 3070 Assignment #2
CheukHim Calvin, LO #100514352

A. Graph algorithms. You may find the following Wikipedia article useful:

https://en.wikipedia.org/wiki/Distance_(graph_theory)

1. [Writing] What is the eccentricity of a vertex in a graph? Illustrate with an example.

Eccentricity of a vertex in a graph is the maximum distance between one vertex and any other vertex. For example, in the graph 2 ------- 1 ------- 3, the eccentricity of 1 is 1, the eccentricity of 2 is 2 and the eccentricity of 3 is 2.

2. [Writing] What is the radius of a graph? Illustrate with an example.

Radius is the minimum eccentricity. For example, in the graph 2 ------- 1 ------- 3. the radius is the eccentricity of 1 which is 1.

3. [Writing] What is the diameter of a graph? Illustrate with an example.

Diameter is the maximum eccentricity. For example, in the graph 2 ------- 1 ------- 3. the radius is the eccentricity of 2 which is 2 and the eccentricity of 3 which 2 as well.

(10 marks)

4. [Algorithm Design] Provide detailed algorithmic solution to compute the *three* properties. You may use functions such as BFS, DFS, and Dijkstra.

```
input: G

for int I = 0 to number of nodes
    // find the shortest path by BFS
    for int s = 0 to number of nodes
        bfs = BFS(G,s)
    end
    // find the eccentricity of s
   if bfs.p[i] != null:
       E[i] = max(bfs.d)
 end
```

```
// for radius
min = E[0]
for i = 0 to number of nodes
        if e[i] < min then min = e[i]
end
Radius = min

// for diameter
max = E[0]
for i = 0 to number of nodes
        if e[i] > max then max = e[i]
end
diameter = max
```

5. [Complexity Analysis] What are the computational complexities of the *three* solutions?

   The complexity of compute eccentricity is O(|V| + |E| + |V|).
   The complexity of compute radius is O(|V|).
   The complexity of compute diameter is O(|V|).

(30 marks)

B. You are given a data file containing pairs of the names of UOIT faculty instructors. It can be obtained at:

http://db.science.uoit.ca/share/instructor-pair.txt

Each pair has shared at least one course in common since 2014.

From the file, construct a *bidirectional* graph. The vertices are the instructors, and an edge exists between *x* and *y* if they have shared a course. Namely, either (*x*, *y*) appears in the file or (*y*, *x*) appears in the file.

Create the adjacency matrix of the graph *G*. You must short the adjacency list alphabetically.

1. [Program] Use BFS to list all the instructors which are connected (directly or indirectly) to “Ken Pu”.
2. [Program] Use DFS to list all the instructors which are connected to “Ken Pu”.
3. [Program & Writing] How many connected components are there? With the help of Google, describe what each connected component represents.

   There are 19 connect components. There are three types of connected components. 1) reflexive: length 0 from any vertex to itself. 2) symmetric: path from one node u to any others node v with the same edges from v to u. 3) transitive: a oath from one node u to any other nodes v and a path from v to any other nodes w.

4. [Program & Writing] For each connected component, measure the number of vertices and the radius of each component. Tabulate your results. Can you gain some insight into the components?

#	V	R
0	152	11
1	251	14
2	9	4
3	16	5
4	2	1
5	2	1
6	5	3
7	2	1
8	3	1
9	3	1
10	5	3
11	2	1
12	6	4
13	2	1
14	2	1
15	5	2

16	4	2
17	2	1
18	2	1

When there are more vertices, the radius is larger.

(60 marks)

Submission:

You must submit a directory with the following structure

```
.
├── report.pdf
└── src
    ├── B1.java
    ├── B2.java
    ├── B3.java
    ├── B4.java
    └── Makefile
```

If you use Python, all *.java files should be *.py files respectively. You are free to include other files for the sake of code organization.

You **must** include a Makefile with at least the following targets defined:

make B1
make B2
make B3
make B4
make clean