

You are asked to use two approaches, *Breath First Search* and *Depth First Search*, to find all connected components of undirected graphs in a given file.

Program requirement: The name of the program should be graphcc.java and I will compile and run your program on our Linux server as follows.

```
javac graphcc.java
java graphcc udGraphs.txt
```

where udGraphs.txt is the name of the input file. If your program fails to compile, you will get 0 point. I may test your program on a different graph file.

Input: The format of the input file is similar to the previous assignments' inputs files except the graphs are undirected. Consider the following example, which is the beginning of file unGraphs.txt and the first graph: **Note:** since undirected graphs must be symmetric, if vertex x and vertex y is directly connected, only one of (x, y) and (y, x) is shown in the edge section.

20 undirected graphs in udGraphs.txt.

**** Note:** Since (x,y) in E implies (y,x) in E , only one edge is listed.

**** G1:** $|V|=10$ $V=\{0,1,...,9\}$

(u, v) $E = \{$
 $(0, 2)$
 $(0, 7)$
 $(2, 8)$
 $(3, 6)$
 $(3, 8)$
 $(4, 5)$
 $(6, 9) \}$

**** G2:** $|V|=10$ $V=\{0,1,...,9\}$

(u, v) $E = \{$
 $(0, 7)$
 $(1, 5)$
 $(2, 5)$
 $(3, 6)$
 $(7, 8) \}$

Output: Both approaches should give the same results, but the order of visiting vertices may be different. You should list all vertices in one component in one line according to the order of visits. **Special requirement,**

during the course of computation, if there are more than one choice to proceed, always select the vertex with a small index. With this requirement, the order of visits will be unique for each graph. For example, the output should be as follows: (check graphs G1 and G2 and explain the results)

Connected components of graphs in udGraphs.txt

** G1's connected components:

Breadth First Search:

0 2 7 8 3 6 9

1

4 5

Depth First Search:

0 2 8 3 6 9 7

1

4 5

** G2's connected components:

Breadth First Search:

0 7 8

1 5 2

3 6

4

9

Depth First Search:

0 7 8

1 5 2

3 6

4

9