

IT 427, Design and Analysis of Algorithms

Programming Assignment 2: Search for connected components

Due date: Sep. 19, 2023, Tuesday, 11:55 PM

50 points (30 on programs, 20 on report)

For this assignment, 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, where `udGraphs.txt` is the name of the input file.

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

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 for this assignment. Consider the following two examples at the beginning of file `unGraphs.txt`. Since undirected graphs must be symmetric, if vertices x and y are adjacent, 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:** $|V|=10$ $V=\{0,1,\dots,9\}$

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

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

```
(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 with this

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 ask yourself why we have the order of visiting)

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

.....

Asg 2 by [your full name]

Note that, after all answers are printed, you should claim ***** Asg 2 by [your full name]** at the end.

Special requirement on the program: Do not create any external class for this project. Code all needed classes and methods in one program file.

Prepare your programs on Linux Server: This is similar to the previous assignment.

- Make a directory **asg2** under your IT427, i.e., `~/IT427/asg2/`. All of your programs and needed files for this assignment should be saved under your `~/IT427/asg2` before run `submit427.sh`.
- Check the contents of my `/home/ad.ilstu.edu/cli2/Public/IT427/asg2` and copy `udGraphs.txt` to your own `~/IT427/asg2`.

There are two parts of submission: Programs (60%) and Reports (40%)

Submission details are same as the previous assignment. Run the submission script with the submission number changed to 2, but you can use the same secret name as follow:

```
bash /home/ad.ilstu.edu/cli2/Public/IT427/submit427.sh peekapoo 2
```

Note that, since I will keep updating `submit427.sh` for different assignment, you have to run the script from my `/home/ad.ilstu.edu/cli2/Public/IT427/` directly for the most recent updated version, i.e., don't copy it to your own directory. Also, once you run `submit427.sh`, don't make any change on the target directory, including compile and run your programs. That will mess up the permission. You will lose significant points if you fail to follow the instructions.

★ Any plagiarism will receive 0 and be reported to the school ★**1. Programs: 30 points. Submission on Linux server.**

The score is based on the correctness and the **programming style, which includes efficiency, appropriateness of data structures, and documentation of your programs**. At the beginning of every program file, put a section of comments including (1) your full name, (2) student ID, (3) a pledge of honesty that you do not copy/modify from other's codes and (4) a declaration of your copyright that no one else should copy/modify the codes. You will receive:

- (a) 30 points: No error with a good programming style.
- (b) 25 ~ 29 points: Minor error and fair programming style.
- (c) 20 ~ 24 points: Some error and not so good but acceptable programming style.
- (d) 15 ~ 19 points: Too many error and bad programming style, but meaningful.
- (e) 10 ~ 15 points: Compilable but not working, provided the program must show reasonable trying.
- (f) 0 points : Fail to meet any of above or plagiarism involved.

2. Report: 20 points. Submission through Canvas.

You have to write up a report and prepare it in pdf format. You don't have to put program output on the report because I will run and exam your program directly on some different input files. You will receive:

- (a) 5 points on the cover page, and
- (b) 15 points on the program style, summary of the methods, algorithms and data structures, and efficiency analysis on time and space in details in terms of big-O notations. If there is any difficulties encountered in this assignment, you can report it.