

# IT 427, Design and Analysis of Algorithms

## Programming Assignment 7: Maximum Flows

**Due date:** Nov. 17, 2023, Friday, 11:55 PM

80 points (60 on programs, 20 on report)

---

For this assignment, you are asked to implement *Ford-Fulkerson's* algorithm and *preflow-push* algorithm in two programs, `FordFulkerson.java` and `preflowPush.java`, respectively.

**Extra 10 points:** We do not require the flow network to be acyclic (i.e., a directed graph without cycle). What is the effect of cycles on the two algorithms, *Ford-Fulkerson's* algorithm and *preflow-push* algorithm, respectively? Does this affect the correctness of the algorithms? If yes, how can it be prevented; if no, explain your reasoning. Discuss this for the extra points in a separate section in your report. (The total points with extra points for this assignment will not be higher than 80 points).

### Prepare your programs on Linux Server:

- Make your directory `~/IT427/asg7/` and all programs and needed files should be saved under this directory before run `submit427.sh` for submission.
- Check the contents of my `/home/ad.ilstu.edu/cli2/Public/IT427/asg7`. There are input files that contain directed weighted graphs as flow networks. We will fixed  $v_0$  as the source and  $v_{n-1}$  as the target where  $|V| = n$ .

**Program requirement:** Your programs will be compiled and run on our Linux server as follows:

```
javac FordFulkerson.java
java FordFulkerson flowNetworks1.txt

javac preflowPush.java
java preflowPush flowNetworks1.txt
```

**Input:** The input file is similar to the previously used weighted directed graphs, except that the number of the graphs in the file is given in the first line and the weights(capacity) of edges are integers and must be treated as integers. Consider `flowNetworks1.txt` as an example in the following:

20 flow networks in flowNetworks1.txt.

```
** G1:  |V|=6, V={0..5}
(u, v, weight) E = {
    ( 0, 3, 53)
    ( 1, 2, 1)
    ( 2, 1, 46)
    ( 2, 5, 3)
    ( 3, 4, 43)
    ( 3, 5, 18)
    ( 4, 1, 20)
    ( 4, 3, 80) }
```

-----

```
** G2:  |V|=6, V=0..5
```

```
.....
```

**Output:** The output of your program should be formatted as follows. The first line has to indicate which algorithm is used. For the flow network with vertex number equal to or less than 10 such as **G1** in the following, the flow network and maximum flow found by the algorithm must be printed on the screen in a matrix with proper alignment as shown in the following. For those with more than 10 vertices, just print the value of the max flow.

Preflow-Push algorithm

```

** G1:  |V|=6
  Flow network:
        0:  1:  2:  3:  4:  5:
        -----
0:      -   -   -   53   -   -
1:      -   -   1   -   -   -
2:      -  46   -   -   -   3
3:      -   -   -   -  43  18
4:      -  20   -  80   -   -
5:      -   -   -   -   -   -

  Maximum flow:
        0:  1:  2:  3:  4:  5:
        -----
0:      -   -   -   19   -   -
1:      -   -   1   -   -   -
2:      -   -   -   -   -   1
3:      -   -   -   -   1  18
4:      -   1   -   -   -   -
5:      -   -   -   -   -   -

Max flow ==> 19 (0 ms)

```

```

** G2:  |V|=6
  Flow network:
  ...
  ...
** G11:  |V|=14
  Max flow ==> 144 (2 ms)

** G12:  |V|=14
  Max flow ==> 193 (0 ms)

  ...
  ...
** G20:  |V|=18
  Max flow ==> 192 (0 ms)

```

by Chung-Chih Li

Needless to say, if both programs are correct, the value of the maximum flow of the same flow network must be the same, **but the flow may be different** as the two algorithms are based on different principles

and may direct flow differently.

Here are some sample output for `java preflowPush flowNetworks4.txt`

```
Preflow-Push algorithm

** G1:  |V|=1000
    Max flow ==> 4793 (420 ms)
    ...
    ...
** G4:  |V|=1000
    Max flow ==> 5125 (386 ms)

by Chung-Chih Li
```

**Submission:** Programs (60 points) and Reports (30 points)

### ★ Any plagiarism will receive 0 and be reported to the school ★

1. There are two program, each worth 30 points. Submission details are same as the previous assignment. Run the submission script with the submission number changed to 7, but you can use the same secret name as follow:

```
bash /home/ad.ilstu.edu/cli2/Public/IT427/submit427.sh peekapoo 7
```

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) 10 ~ 19 points: Too many error and bad programming style, but meaningful.
  - (e) 5 ~ 9 points: Compilable but not working and the program must show reasonable trying.
  - (f) 0 points : Fail to meet any of aforementioned qualities 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 as I will run and exam your program directly on some different input files.
    - (a) 5 points on the cover page, and
    - (b) 15 points on algorithm descriptions, summary of the methods, 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. If your analysis is not clearly related to your program and provides sufficient justification, your report score will not be higher than 50%.