Faculty of Science

IT Departement

TP GRAF – Maximum Flow computing report

Author:
TCHOULAK Ramzi
BENAKMOUME Yacine

17/12/2020

1 Introduction

1.1 Presentation

A flow network is a graph where the weight of an edge indicates its capacity. We were asked to make a program that computes the maximum flow problem by implementing the Ford-Fulkerson algorithm to calculate the residual graph and the augmenting paths.

1.2 Project Goals

1. Create a graph
2. Manage the graph by adding its components.
3. Create a Flow Network from an imported dot file.
4. Compute and solve the problems of the maximum flow using Ford Fulkerson algorithm.

2 Implementation phase

2.1 Project files

1. **Graf.jar** : This file represents the library that we had implemented on the PW2 subject, it contains the API that makes us able to create and manipulate the graphs. In our project, PW3, we use it to create the Flow Network that describes the augmenting paths in each iteration, also the maximum flow of a graph and the residuals graphs.
2. **Interface.java** : The Interface proposed to the users to manipulate the Graphs (PW2)
3. **Part2.java** : The Interface proposed to the users to manipulate the Flow Network
4. **FlowNetwork.java** : Contains the class that manipulates the Flow Network with all the functions and the methods, it is the API for our project (PW3), it contains the graph that uses the Graf library, the Ford Fulkerson Algorithm implementation for calculating the maximum flow , the functions get the startNode and the endNode, the files for reading and writing the DOT

files of the flow problem and also the manipulation of the residual graph (it's kind of a flow network) and the calculation of the augmenting path with **BFS** (we have add new BFS function that we send in its parameter the start and end Node of the Flow.

5. **Makefile** : The Makefile that makes us able to execute the project and its dependencies.

2.2 Running the project

The project is in the ZIP format, to begin, you must extract all the files from the archive

1. **Method 1** : Run your terminal in the directory that contains the make file, then run the command make, and IHM should appear and gives you the menu to use.
2. **Method 2** : You can use Intellij idea or Eclipse and import the project and run the project.

3 Principles

The implementation of Ford Fulkerson Algorithm used in our project is called Edmonds-Karp Algorithm. The idea of Edmonds-Karp is to use BFS in Ford Fulkerson implementation as BFS always picks a path with minimum number of edges.

3.1 Complexity

The worst case time complexity can be reduced to $\theta(\mathrm{V}.E^2)$. Our used implementation uses adjacency matrix representation though where BFS takes $\theta(V^2)$ time, the time complexity of our algorithm implementation is $\theta(\mathrm{E}.V^3)$

3.2 Demonstration and Display

Swing is the used Framework for GUI display.

Here's an example of an execution

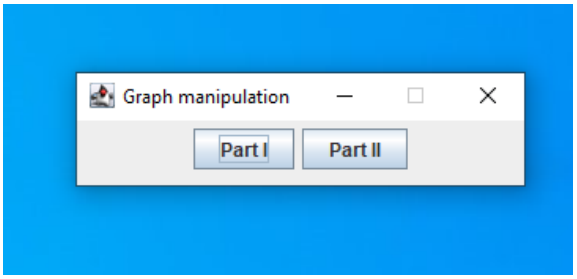


Figure 1: Part selection PW2 and PW3

After selecting Part ll which means PW3 project we import the DOT file as the following

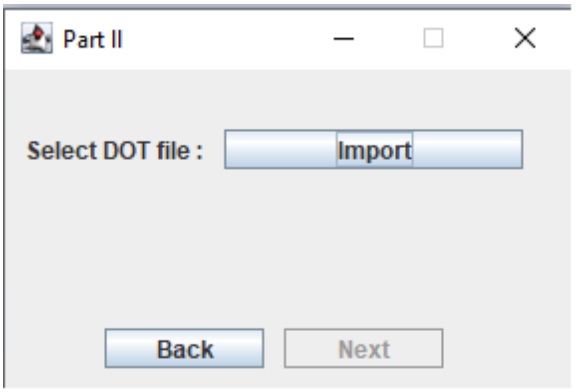


Figure 2: Interface and DOT file import

Once the file is imported, the current DOT file is parsed and converted to a graph and displayed as the figure below

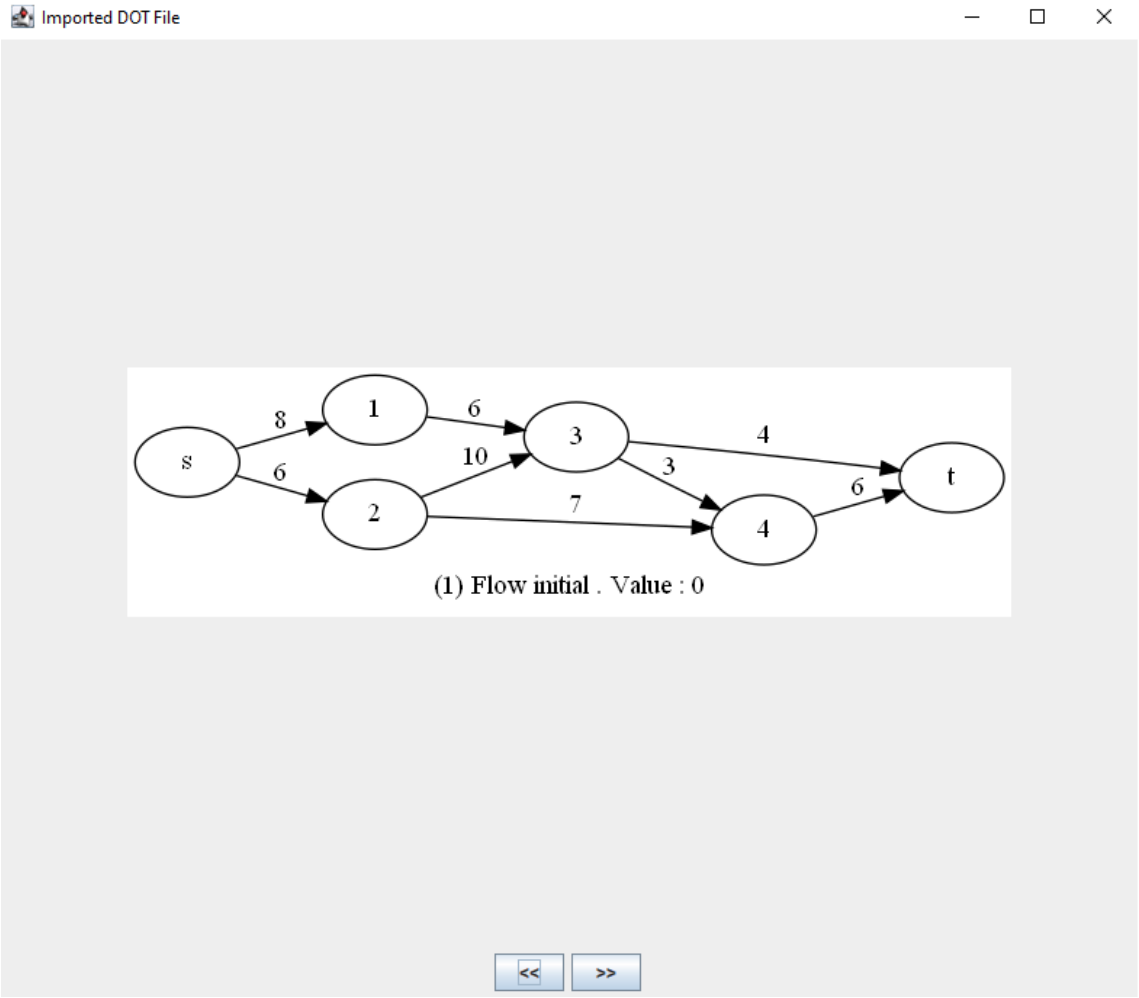


Figure 3: Imported graph display

After sliding right this is one of the states of Floyd Fulkerson transformation

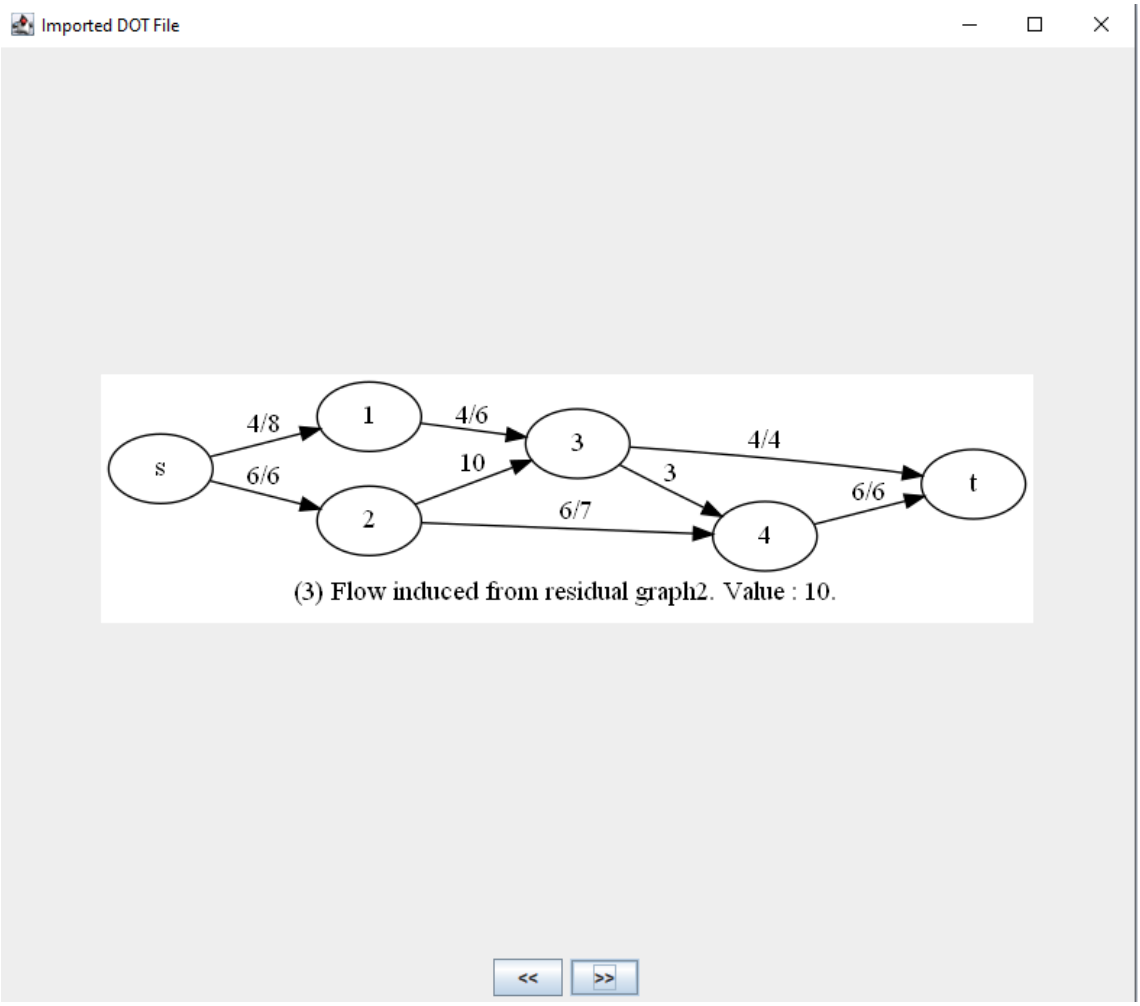


Figure 4: Step 03 of Floyd Fulkerson - Flow Network

Final step, once the algorithm is over the figure below is displayed

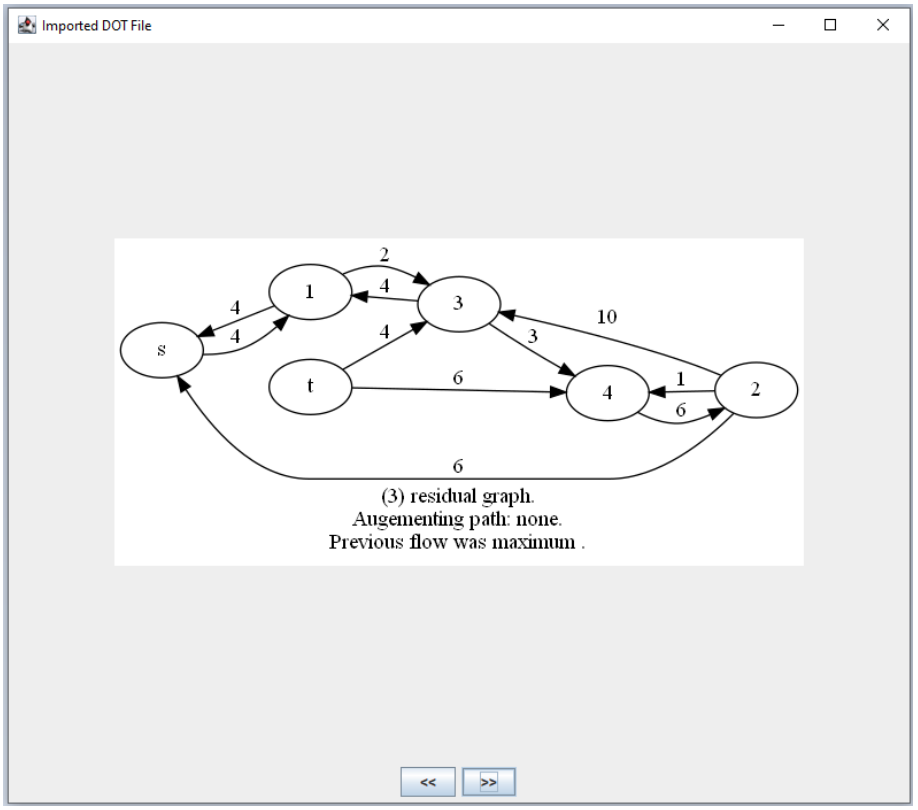
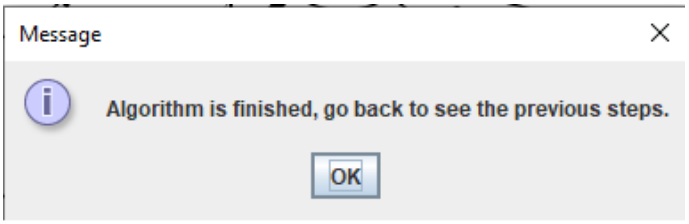


Figure 5: Last step of the algorihm - nothing to add

Algorithm finished alert



References

[1] Support given : Ford Fulkerson algorithm and Maximum Flow

[2] Add-on resource - Ford-Fulkerson Algorithm for Maximum Flow Problem GfG Course - GeeksForGeeks