PW4 Project: Maximum Flow

Evaluated project to be presented as a demo during the last PW session

Work to be done in pairs

Foreword of PW2 also applies for PW4.

Subject: Computing the maximum flow between a source and a sink in a flow network modelled as a weighted graph G

This project aims at implementing the Ford-Fulkerson method for computing the maximum flow of a flow network. The problem itself as well as a solution to it have been previously studied during a Lecture Course, so they are assumed to be known and understood. Consequently this subject only gives some general implementation instructions.

Let $G = (V, E)$ be a weighted oriented graph modelling a flow network. It can for example describe a road network, or a water network, etc. If for example G models a road network, the weight of each edge models its *capacity*, for example the maximum number of vehicles allowed in a day.

The aim of this work is to compute the maximum flow that the network is able to support for shipping items between two particular nodes of G: a source s and a sink (or target) t.

Work to be done

You will use the graph library you have developed for PW2 to implement this project. You will possibly have to adapt it to the purpose of the project, for example for associating each edge with a weight (= a capacity) if this is not already done.

You will apply the Ford-Fulkerson method, as studied in Lecture Courses, to solve the problem. Several kind of graphs will have to be manipulated: the original flow network itself, as well as the successive flows and residual networks. As suggested during the Lectures, the flow of a graph can be modelled as a bi-dimensional array (a matrix) where $f[u, v]$ gives the flow value going from node u to node v. The successive residual networks G_f will be stored thanks to your PW2 graph library, similarly as if they were flow networks.

In the beginning, the flow network is initialized with a null flow value on each edge. While there is an augmenting path between source s and sink t in the residual network G_f, then the flow values of the edges along the augmenting path are increased accordingly. At each step the residual network is updated to take into account the modified residual capacities of the edges. The maximum flow is reached when no more augmenting path can be found in the residual network.

The efficiency of the method depends on how the augmenting path is computed. You are asked to propose and implement several ways (heuristics) for computing this augmenting path.

Program and format

Your program will offer the same menu as PW2, but augmented with new commands for selecting the source and sink states, computing and displaying the maximum flow between two nodes of the graph, etc.

As an input, you will read a file describing the flow network such as:

```
# maximum number of nodes
11
# node : [neighbour/capacity]
1 : 4/2, 7/3, 3/4
2 : 5/12, 7/3
3 : 1/2, 6/4
4 : 7/3, 1/5, 6/2
5 : 2/8, 8/2
6 : 4/10, 3/1
7 : 2/2, 4/7, 1/3
8 : 5/3
```

As stated before, implementing this project requires using the graph library you have developed for the purpose of PW2. In case your own PW2 library is incomplete or has bugs, you are allowed to use the library of someone else in your promotion, but you have to explicitly say so to your teacher and then in your final report.

Work submission

Submit your work on Moodle by your last PW session. The precise date will be displayed on Moodle. Your submission should contain: your source files, the example graphs you have tested your program with, a README file as well as a report. The README briefly describes how to compile, launch and use your program, and lists its files and what they aim at. The report explains what heuristics (if so) you have proposed and implemented, and may contain any additional information you'd find useful to be known.