BLG 336E – ANALYSIS OF ALGORITHMS II

Project 3

Muhammed Burak Buğrul
150140015

Introduction

We are wanted to implement Ford_fulkerson algorithm in order to solve a reviewer-publication matchin problem.

Environment

Project implemented in an **Ubuntu 16** machine with **C++**.

Code

In the code there are some explanations(comments).

Analysis

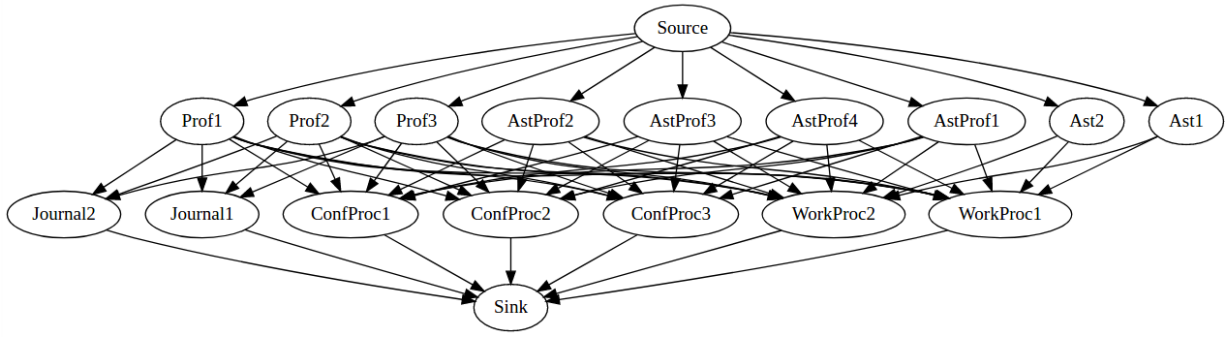
- Some matching problems can be solved by network-flow algorithms if they are suitable. This problem is one of them. Every flow represents a match, and with these flows given limitations(time limits of reviewers, title limits for a publication etc.) can be satisfied if they can placed to weights of edges properly. In this problem:

  - Giving review limits as weights to edges between reviewers and source provides review limits on flow. But amount of every flow should be 1, because every flow means a reviewer-publication match.

  - Giving 1 as a weight to every potential reviewer-publication edge provides amounts of every flow to be 1 and inhibits a reviewer from review a publications more than once.

  - Giving required review number of publications as weight to edge between publication and sink provides required review numbers for publications. Namely, it limits review number of a publication.

- Visualization of graph given in input(Edge eights are not given for the sake of simplicity):



http://www.webgraphviz.com/ for graph visualization

- In the code, graph represented with a VxV matrice(let V be the total number of nodes). And in that graph **Ford-Fulkerson** algorithm performed. Time complexity of **BFS** is **$O(V^2)$** because of the graph representation. An there can be **$O(V^2)$** reviewer-publication matches. Every BFS call finds one match so time complexity of the algorithm is **$O(V^4)$**.