TCSS 543B: Advanced Algorithms
Empirical study project on network flow
Slides Due: Tuesday Dec 3, 2019 at 1:30 pm
Code and Report Due: Friday Dec 6, 2019 at 11:59 pm

1 Overview

Problem description In this project, you will be conducting an empirical study to see which of the network flow algorithms is better than the others. It is unlikely that you will discover that one algorithm is better than the others for all inputs. In fact, our goal in this project is to see if you can figure out for what kinds of graphs one algorithm does better than the others. The algorithms you will implement and run experiments on are: the Ford-Fulkerson algorithm (Ch 7.1), the scaling Ford-Fulkerson algorithm (Ch 7.3), and the preflow-push algorithm (Ch 7.4). You are expected to implement the algorithms yourself, as opposed to using existing code that you might have found online.

Logistics Each project team should have 3-4 members. Note that I reserve the right to assign different grades to students in the same team based on the division of labor. The project counts for 25% of your final grade. The project will be graded out of a total of 25 points.

- In-class presentation and demo: 7 points
- Code and documentation: 8 points
- Report: 10 points

Code Graph code and some code to generate graphs is provided on Canvas. You should not make changes to the graph code. The reason for this is so that the underlying graph code is all the same, which allows us to make fair comparisons among different groups.

Input graphs You might consider the following when generating input graphs:

1. Random graphs (each edge is generated with probability p).
2. Mesh graphs.
3. Bipartite graphs.
4. The range of capacities on the edges.

2 Timeline

Tue Nov 26 Submit group membership. Discussion with your team and the instructor in class. Informal project update and feedback opportunity.

Tue Dec 3, 1:30 pm Slides for each team due via Canvas. One submission per group please.

Tue Dec 3 and Thur Dec 5, during class Every team gives a final project presentation and a demo of the implementation (15 minutes).

Fri Dec 6, midnight Your group report and code are due via Canvas. One submission per group please.

3 Deliverables

Slides (1 upload per team; Dec 3, 1:30 pm) Upload your slides on Canvas.

Code (1 upload per team; Dec 6, midnight) To be submitted on the course Canvas. Your network flow algorithms must be implemented in Java. You must submit a `zip` file containing the following:

- All java files and makefile needed to compile your code.
- Compiled `.class` files.
- Generate input graphs and submit your input testing files.
- Submit an additional documentation file in PDF. Specifically, your documentation should include the structure of your code. Name each routine and describe in 1-2 sentences what each routine does. This file should also show the output of your code using each of your input testing files.
- You must name your source code containing the `main` method **tcss543.java**. We will execute your code using the following command: `java tcss543 < input.txt`
- Your implementation should output the value of the flow of the input graphs using each of the three methods.

In addition to your own input testing files, we will test your implementation using additional input files. Points will be deducted if your submission does not follow the above specifications.

Group report (1 upload per team; Dec 6, midnight) In addition to your code for your empirical study, you are expected to write a report on your work. The submitted report should be typeset using any common software and submitted as a PDF. The report is not quite an academic paper, but it follows roughly the same format. Your report should be divided into sections as follows:

- Introduction: A description of the problem (network flow), what the goals of the study are, and a brief description of the results.
- Methodology: A description of the code, what algorithms you implemented, implementation difficulties, etc. You should also describe how you tested your code, how you generated your input instances (e.g., how you generated your graphs), etc. For example, for network flow, how did you choose your s and t? You should also say something about how many times you ran your algorithm on input instances, what range of graph sizes (number of vertices and number of edges) you used, and how you systematically investigated the algorithms.
- Results: What did you discover about your algorithms? Which was faster? Was one or more algorithm more sensitive to the number of edges in the graph, the number of vertices in the graph (or in network flow, the value C)?
- Future work: If you were to do the experiments over again, what would you do differently? What future experiments can you think of that might be interesting, given what you learned from this project? Are there any applications or data that you would be interested to apply your implementation to?

- Division of labor: Describe the parts of the project each team member worked on (which algorithm(s) you implemented, which input graphs you tested, who you worked with on what parts, etc.) and what parts of the project your teammates worked on.
- Lessons learned: What you learned from the project.

Formatting guidelines: up to 6 pages, double column, ACM Proceedings format.[1] In case you need more than 6 pages, consider splitting your material in a main paper and an appendix.

[1] http://www.acm.org/sigs/publications/proceedings-templates