Advanced Graph Algorithms and Optimization	Spring 2023

Classical Algorithms for Maximum Flow

R. Kyng & M. Probst Gutenberg — *Problem Set 7 — Thursday, April 27th*

The exercises for this week will not count toward your grade, but you are highly encouraged to solve them all. This exercise sheet has exercises related to week 7. We encourage you to start early so you have time to go through everything.

To get feedback, you must hand in your solutions by 23:59 on May 4. Both hand-written and LaTeX solutions are acceptable, but we will only attempt to read legible text.

Exercise 1: Implementing Field Preservations for Cut-Link Tree Rotations

In Chapter 14, Section 14.3, we described the operations $\textsc{PLink}(u,v)$ and $\textsc{PCut}(u,v)$ by doing $O(\log n)$ tree rotations (in expectation). However, we omitted the details. Here we ask you to give the pseudo-code for the operation $\textsc{PTreeRotation}(v,w)$ where it is assumed that on input v is the parent of w and the operation manipulates the tree over the path such that v and w change position as described in the script. For simplicity, you are allowed to assume that w is the left child of v, and that the nodes $\mathit{left}_{\mathcal{P}}(v), \mathit{right}_{\mathcal{P}}(v), \mathit{left}_{\mathcal{P}}(w), \mathit{right}_{\mathcal{P}}(w)$ exist. Accompany your pseudo-code with a brief analysis that confirms that the run-time is indeed $O(1)$.

Exercise 2: Max Flow in directed Graphs with Edge Capacities

Consider directed graph $G = (V, E, \boldsymbol{c})$ with arbitrary capacities $\boldsymbol{c} \geq \mathbf{0}$.

Let $\boldsymbol{B} \in \mathbb{R}^{E\times V}$ be the edge vertex incidence matrix of the graph, i.e. if $e \in E$ and $(u,v) = e$ then $\boldsymbol{B}(e,u) = 1$ and $\boldsymbol{B}(e,v) = -1$.

We let $\boldsymbol{\chi}_v \in \mathbb{R}^V$ denote the indicator of vertex v, i.e. $\boldsymbol{\chi}_v(v) = 1$ and $\boldsymbol{\chi}_v(u) = 0$ for $u \neq v$.

We let $s \in V$ denote the flow "source" and $t \in V$ the flow "sink".

The maximum flow problem is given by

$$\begin{aligned} \max_{\boldsymbol{f}\in\mathbb{R}^E, F\geq 0} \quad & F \\ \text{s.t. } \boldsymbol{B}\boldsymbol{f} &= F(-\boldsymbol{\chi}_s + \boldsymbol{\chi}_t) \\ \mathbf{0} \leq \boldsymbol{f} &\leq \boldsymbol{c} \end{aligned}$$

In the context of a given maximum flow problem, for a flow $\boldsymbol{f}$ satisfying $\boldsymbol{B}\boldsymbol{f} = F(-\boldsymbol{\chi}_s + \boldsymbol{\chi}_t)$, we define $\mathrm{val}(\boldsymbol{f}) = F$.

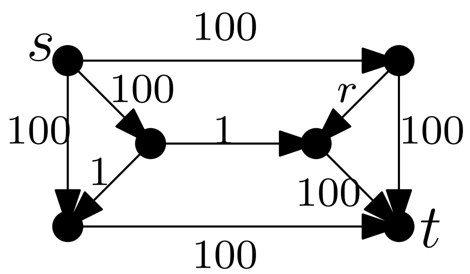
Let $\boldsymbol{f}^*$ denote a feasible flow maximizing F, so that the maximum attainable flow value F is $\mathrm{val}(\boldsymbol{f}^*)$.

Exercise 2.A: Convergence of Ford-Fulkerson

Show that the Ford-Fulkerson algorithm may not terminate; moreover, it may converge toward a value not equal to the value of the maximum flow.

Hint: You might use the graph below with the given capacities, where $r = \frac{\sqrt{5}-1}{2}$ *(which implies that* $r^2 = 1 - r$*).*

Warning: This exercise is quite unpleasant – you may want to focus on the other ones, and only do this one if you have time to spare.



Exercise 2.B: Iterative Refinement for Maximum Flow

Suppose we have an algorithm FlowRefine, which given a maximum flow instance $G = (V, E, \boldsymbol{c})$ with source $s \in V$ and sink $t \in V$ returns a feasible s-t flow $\tilde{\boldsymbol{f}}$, i.e. $\boldsymbol{B}\tilde{\boldsymbol{f}} = F(-\boldsymbol{\chi}_s + \boldsymbol{\chi}_t)$ for some F, and $\boldsymbol{0} \leq \tilde{\boldsymbol{f}} \leq \boldsymbol{c}$, and $\tilde{\boldsymbol{f}}$ is guaranteed to route at least half the maximum flow, i.e. $F = \mathrm{val}(\tilde{\boldsymbol{f}}) \geq 0.5\,\mathrm{val}(\boldsymbol{f}^*)$.

Suppose that the running time of FlowRefine is $O(|E|^c)$ for some constant $c \geq 1$.

Explain how we can use FlowRefine to find a flow $\hat{\boldsymbol{f}}$ that routes at least $(1 - \epsilon)\,\mathrm{val}(\boldsymbol{f}^*)$ in time $O(|E|^c \log(1/\epsilon))$.