Technische Universität München
Fakultät für Informatik
Lehrstuhl für Algorithmen und Komplexität
Prof. Dr. Harald Räcke
Richard Stotz

Winter term 2019/20
Assignment 13
January 20, 2020

Efficient Algorithms and Data Structures I

Deadline: January 27, 10:15 am in the ***Efficient Algorithms*** *mailbox.*

Homework 1 (4 Points)

Prove the following statements about maximum flow networks.

(a) If all edge capacities are even integers, then the maximum flow value is an even integer.

(b) Let e be an edge that belongs to some minimum cut. Show that any maximum flow saturates the edge e.

Homework 2 (5 Points)

Let $G = (V, E)$ be a network with two vertices $s, t \in V$. We call two $s-t$-paths edge-disjoint if they do not share an edge. Two $s-t$-paths are vertex-disjoint if they have no vertices in common other than s and t.
Prove or disprove the following statements

(a) There are $k > 1$ pairwise edge-disjoint $s-t$-paths in the network if and only if after deleting any $k-1$ edges, there still exists a path from s to t.

(b) There are $k > 1$ pairwise vertex-disjoint $s-t$-paths in the network if and only if after deleting any $k-1$ vertices, there still exists a path from s to t.

Homework 3 (5 Points)

We say that a bipartite graph $G = (V, E)$, where $V = L \cup R$, is d-regular if every vertex $v \in V$ has degree exactly d. Every d-regular bipartite graph has $|L| = |R|$. Prove that every d-regular bipartite graph has a matching of cardinality $|L|$ by arguing that a minimum cut of the corresponding flow network has capacity $|L|$.

Homework 4 (6 Points)

An unknown attacker has targeted a flow network of n nodes, source s and target t and unit capacities on all edges. The attacker has separated nodes s and t by removing k edges, where k is the capacity of a minimum cut separating s and t.
Detective Dirk wants to find out which edges have been removed by the attacker. He knows how the network looked like before the attack, but he does not know how it looks like after the attack. However, Dirk can ping vertices in the damaged network. A ping for vertex v is successful, if the damaged network still contains a path from s to v.

Dirk wants to work as fast as possible, so he aims to minimize the number of pings.

(a) Suppose that the network is a simple path from s to t. Show how $\mathcal{O}(\log n)$ pings are sufficient to find the damaged edge.

(b) Dirk conjectures that there exists an algorithm needing only $\mathcal{O}(k \log n)$ pings in an arbitrary network. Show that Dirk is right by giving such an algorithm. Prove correctness of your algorithm and prove that it reaches the required ping bound.

Tutorial Exercise 1

A shipping company wants to phase out a fleet of s (homogeneous) cargo ships over a period of p years. Its objective is to maximize its cash assets at the end of the p years by considering the possibility of prematurely selling ships and temporarily replacing them by charter ships.
The company faces a known nonincreasing demand for ships. Let d_k denote the demand of ships in year k. Each ship earns a revenue of r_k units in period k. At the beginning of year k, the company can sell any ship that it owns, accruing a cash inflow of s_k dollars. If the company does not own sufficiently many ships to meet its demand, it must hire additional charter ships. Let h_k denote the cost of hiring a ship for the kth year.
The shipping company wants to meet its commitments and at the same time maximize the cash assets at the end of the pth year.
Model this problem as a minimum cost flow problem!

Tutorial Exercise 2

Programmer Paul wants to transfer each of his n applications from `Python 7` to `Python 8`. As the new language has a much better performance, transferring application i saves him $b_i \geq 0$ Euros over a year. Compatibility problems, however, make it expensive to move only some of the applications: If only one of applications i and j moves to the new language, the extra cost over a year is $c_{ij} = c_{ji} > 0$ Euros.

1. Unfortunately, Paul finds out that Application 1 cannot be transferred to `Python 8` at all. Using a suitable maximum flow network, decide which applications (if any) should be transferred to `Python 8` in order to maximize Paul's savings. Show that your solution is correct.

2. In a more general scenario, the benefit b_i of moving to the new language might be negative, i.e., moving an application may incur cost. Again, use a suitable maximum flow network to decide which applications (if any) should be transferred. You do not need to show that your algorithm works correctly.

For both parts of the question, describe precisely how your network is constructed.

```
No matter what gets in the way and which way
the wind does blow
And as long as it does I'll just sit here
And watch the river flow
                                   - B. Dylan
```