Prof. Friedrich Eisenbrand
Martin Niemeier

Due Date: May 20, 2010
Discussions: May 06

Discrete Optimization

Spring 2010

Assignment Sheet 5

You can hand in written solutions for up to two of the exercises marked with (*) or (Δ) to obtain bonus points. The duedate for this is May 20, 2010, before the exercise session starts. Math students are restricted to exercises marked with (*). Non-math students can choose between (*) and (Δ) exercises.

Exercise 1
Let $T = (V, A)$ be a tree rooted in $r \in V$. Show that there are no two different paths from r to another node of the tree.

Exercise 2 ($*$)
Let $Q =< u_1, \ldots, u_k >$ be the queue before an iteration of the **while** loop of the breadth-first-search algorithm. Show that $D[u_i]$ is monotonously increasing and that $D[u_1] + 1 \geq D[u_k]$. Conclude that the sequence of assigned labels (over time) is a monotonously increasing sequence.

Exercise 3
There are n types of animals, and you want to assign them to two stables. Unfortunately, some animals would eat other animals when left unattended. Therefore you need to assign the animals carefully. There are m relations of the form "u eats v", where u and v are animals.

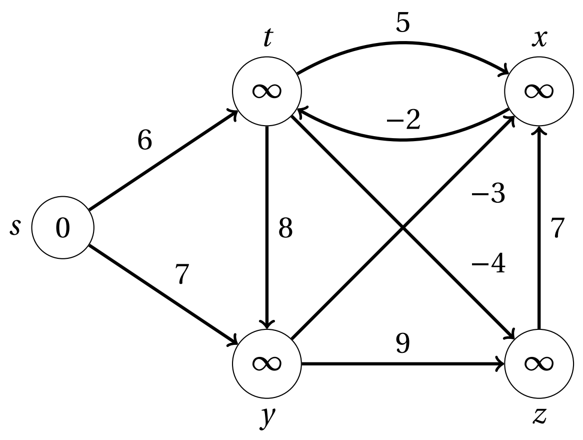
Find an $O(n + m)$ algorithm that decides whether there is an assignment of animals to the two stables such that no animal eats another one of the same stable, and outputs a feasible assignment.

Hint: Breadth-first-search might be useful.

Exercise 4
Consider the directed graph below with distance labels.

Perform the Bellman-Ford algorithm on this graph, using s as source vertex. For each $k = 1, \ldots, 5$: Draw the graph with the f_k values as node labels. For each node, mark an arc for which the minimum of step (ii) is achieved (if any).

The node labels for $k = 0$ are already given in the graph.



Exercise 5 ($*$)
As discussed in the lecture, to compute shortest paths we require that our input graphs do not have cycles of negative length. In this exercise we will show how to detect negative cycles with the Bellman-Ford algorithm.

Let $D = (V, A)$ be a digraph, and $c\colon A \to \mathbb{R}$ a length function. Let $n := |V|$.

1. Let $f_k(v)$ be defined for all $k = 0, \ldots, n$ and $v \in V$ as in the lecture. Show that D contains a cycle of negative length *reachable from the source node s* if and only if $f_n(v) \neq f_{n-1}(v)$ for some $v \in V$.

2. Consider the following modification of the Bellman-Ford algorithm: We maintain a *predecessor* vector $\Pi \in V^V$, i.e. for $v \in V$ we say that $\Pi(v) \in V$ is the *predecessor* of v.

   Initially we set $\Pi(v) :=$ INVALID for all $v \in V$. Now in step (ii) of the algorithm, when calculating $f_{k+1}(v)$ and $f_{k+1}(v) = f_k(u) + c(u, v)$ for some $u \in V$, we set $\Pi(v) := u$.

   Assume that D has a negative cycle reachable from s. Show how to use this modified Bellman-Ford algorithm to find a cycle of negative length.

Exercise 6 (Δ)
Arbitrage is the use of discrepancies in currency exchange rates to transform one unit of currency into more than one unit of the same currency. For example, suppose that 1 CHF buys 46.4 Indian rupees, 1 Indian rupee buys 2.5 Japanese yen, and 1 Japanese yen buys 0.0091 CHF. Then, by converting currencies, a trader can start with 1 CHF and buy $46.4 \times 2.5 \times 0.0091 = 1.0556$ CHF, thus turning a profit of 5.56 percent.

Suppose that we are given n currencies $c_1, c_2, \ldots, c_n$ and an $n \times n$ table R of exchange rates, such that one unit of currency c_i bys $R[i, j]$ units of curency c_j. Given an efficient algorithm to determine whether there exists a sequence of currencies c_{i_1}, c_{i_2}, …, c_{i_k} such that

$$R[i_1, i_2] \cdot R[i_2, i_3] \cdots R[i_{k-1}, i_k] \cdot R[i_k, i_1] > 1,$$

and computes such a sequence if it exists.

Hint: Exercise 5 might help.