TEL AVIV UNIVERSITY
Department of Computer Science
0368.4281 – Advanced topics in algorithms
Spring Semester, 2012/2013

Homework 3, May 7, 2013

Due on May 21. Please put a hardcopy in my mailbox and keep a copy of the homework!

1. Prove that Dinic's algorithm when capacities are all the same, runs in $O(mn^{2/3})$ time. Give precise arguments both for the time it takes to compute a blocking flow, and for the number of blocking flow computations that are required.

2. Assume that we have an acyclic flow f from s to t of some value Γ. Show how to obtain an acyclic flow f' from s to t of value $\Delta < \Gamma$ in linear time.

3. Assume that in some iteration of the algorithm of Goldberg and Rao we computed a blocking flow f' in the admissible subgraph. Prove that after we add the blocking flow f' to the current flow f the distance from s to t in the (new) admissible subgraph is strictly larger than the distance from s to t in the (old) admissible subgraph.

4. In this question we develop in two stages an algorithm that finds a blocking flow from s to t in a directed acyclic graph $G(V, E)$ in $O(m \log(\frac{n^2}{m}))$ time where $n = |V|$ and $m = |E|$.

Each vertex in $V \setminus \{s\}$ is either *blocked* or *unblocked*. Initially all vertices in $V \setminus \{s\}$ are unblocked. We also maintain a preflow f which is initialized by setting $f(s, v) = u(s, v)$ for all edges outgoing from s. The excess $e(v)$ of a vertex v is the difference between the flow incoming to v and the flow outgoing of v and is maintained nonnegative for all v.

A vertex v with $e(v) > 0$ (more flow entering v than outgoing of v) is *active*. We denote by $r(v, w)$ the residual capacity of an arc (v, w) with respect to the current preflow.

The algorithm makes use of the three operations push, pull, and block. A push operation applies to an arc (v, w) such that v is active, v and w are unblocked, and $r(v, w) > 0$. It consists of increasing $f(v, w)$ by $\min\{e(v), r(v, w)\}$. A pull operation applies to an arc (v, w) such that w is active and blocked and $f(v, w) > 0$. It consists of decreasing $f(v, w)$ by $\min\{e(w), f(v, w)\}$. A block operation applies to an unblocked vertex v such that, for every arc (v, w), either $r(v, w) = 0$ or w is blocked. It consists of making v blocked.

A *discharge* operation applies to an active vertex v. The operation either pushes on an arc (v, w), blocks v, or pulls on an arc (u, v), whichever applies, and repeats this until v becomes inactive.

The *first-active blocking flow algorithm* uses a list L to determine the order in which to process active vertices. The list L contains all vertices of G, initially in topological order. The algorithm consists of repeating the following three steps until there are no active vertices: Select the first active vertex on L, say v. Apply a discharge operation to v. If the discharge has made v blocked, move v to the front of L.

a. Prove carefully and formally that this algorithm finds a blocking flow in $O(n^2)$ time.

b. Show how to combine the first-active blocking flow algorithm with dynamic trees and finger search trees to reduce the running time to $O(m \log \frac{n^2}{m})$. Argue formally that this is indeed the running time of your algorithm.