CSCE629 Analysis of Algorithms
Homework 8

Texas A&M U, Fall 2019 — *10/25/19*
Lecturer: Fang Song — *Due: 10am, 11/01/19*

Instructions.

- Typeset your submission by LaTeX, and submit in PDF format. Your solutions will be graded on *correctness* and *clarity*. You should only submit work that you believe to be correct, and you will get significantly more partial credit if you clearly identify the gap(s) in your solution. You may opt for the "I'll take 15%" option.
- You may collaborate with others on this problem set. However, you must **write up your own solutions** and **list your collaborators and any external sources** for each problem. Be ready to explain your solutions orally to a course staff if asked.
- For problems that require you to provide an algorithm, you must give a precise description of the algorithm, together with a proof of correctness and an analysis of its running time. You may use algorithms from class as subroutines. You may also use any facts that we proved in class or from the book.
- **If you describe a Greedy algorithm, you will get no credit without a formal proof of correctness, even if your algorithm is correct.**

This assignment contains 5 questions, 4 pages for the total of 50 points and 0 bonus points. A random subset of the problems will be graded.

Exercises. Do not turn in.

1. Let f and f' be feasible (s,t)-flows in a flow network G, such that $v(f') > v(f)$. Prove that there is a feasible (s,t)-flow with value $v(f') - v(f)$ in the residual network G_f.
2. Let $u \to v$ be an arbitrary edge in an arbitrary flow network G. Prove that if there is a minimum (s,t)-cut (S,T) such that $u \in S$ and $v \in T$, then there is *no* minimum cut (S',T') such that $u \in T'$ and $v \in S'$.

Problems to turn in.

1. (10 points) (Stabbing points) Let X be a set of n intervals on the real line. We say that a set P of points *stabs* X if every interval in X contains at least one point in P. Describe and analyze an efficient algorithm to compute the smallest set of points that stabs X. Assume that your input consists of two arrays $L[1, \ldots, n]$ and $R[1, \ldots, n]$, representing the left and right endpoints of the intervals in X. (N.B. If you use a greedy algorithm, you must prove its correctness.)

2. (10 points) (Maximum spanning tree) Describe and analyze an algorithm to compute the *maximum*-weight spanning tree of a given edge-weighted graph.

3. (Demand) Suppose instead of capacities, we consider networks where each edge $u \to v$ has a non-negative *demand* $d(u \to v)$. Now an (s,t)-flow f is *feasible* if and only if $f(u \to v) \geq d(u \to v)$ for every edge $u \to v$. (Feasible flow values can now be arbitrarily large.) A natural problem in this setting is to find a feasible (s,t)-flow of *minimum* value.

   (a) (10 points) Describe an efficient algorithm to compute a feasible (s,t)-flow, given the graph, the demand function, and vertices s and t as input. (Hint: find a flow that is non-zero everywhere, and then scale it up to make it feasible.)

   (b) (10 points) Suppose you have access to a subroutine MaxFlow that computes *maximum* flows in networks with edge capacities. Describe an efficient algorithm to compute a *minimum* flow in a given network with edge demands; your algorithm should call MaxFlow exactly once.

   (c) (10 points) State and prove an analogue of the max-flow min-cut theorem for this setting. (Do minimum flows correspond to maximum cuts?)