Algorithms — Winter 2007

Problem Set #5 — Instructor: Anna Karlin

Due on **Feb 17, 2007** by 7pm.

Instructions: You are welcome to brainstorm on ideas for solving these problems with fellow students taking the class. You may also collaborate with one other classmate on writing up your solutions. If you do collaborate in any way, please acknowledge for each problem the people you worked with on that problem. Please do not post solution ideas to discussion boards – discussion boards can be used to clarify the problems and perhaps to discuss specific examples. Learning will not be effective if people can find solutions (even partial solutions) to the problems on the dicussion board, on the Web or in other algorithms textbooks.

Most of the problems require only one or two key ideas for their solution – spelling out these ideas should give you most of the credit for the problem even if you err in some finer details. So, make sure you clearly write down the main idea(s) behind your solution even if you could not figure out a complete solution.

Be sure to carefully read the grading guidelines page linked off the course web page.

Readings: Kleinberg and Tardos: Chapters 6, 7.

Each problem is worth 10 points unless noted otherwise. All problem numbers refer to the Kleinberg-Tardos textbook.

1. Chapter 6, Problem 19.

2. State whether the following statements are True of False, and justify your answer.

   (a) Let $G = (V, E)$ be a directed graph. Let $a, b, c \in V$ be three distinct vertices such that in the graph G there exist k mutually edge-disjoint paths from a to b, as well as k mutually edge-disjoint paths from b to c. Then there also exist k mutually edge-disjoint paths between a and c.

   (b) Consider the following "Forward-Edge Only" algorithm for computing s-t flows. The algorithm runs in a sequence of augmentation steps till there is no s-t path in the residual graph, except that we use a variant of the residual graph that *only includes the forward edges*. In other words, the algorithm searches for s-t paths in a graph $\tilde{G}_f$ consisting only of edges e for which $f(e) < c(e)$, and terminates when there is no augmenting path consisting entirely of such edges. Note that we do not prescribe how this algorithm chooses its forward-edge paths, it may choose them in any fashion it wants, provided that it terminates only when there are no forward-edge paths.

   Now to our claim: On every instance of the Maximum Flow problem, the forward-edge only algorithm returns a flow with value at least 1/4 of the maximum-flow value (regardless of how it chooses it forward-edge paths).

3. Chapter 7, Problem 19.

4. Chapter 7, Problem 23.