Algorithm Design and Analysis October 28, 2008
Pennsylvania State University CSE 565, Fall 2008
Adam Smith Handout 12

Homework 9 – Due Monday, November 3, 2008

Please refer to the general information handout for the full homework policy and options.

Reminders

- Your solutions are due before the lecture. Late homework will not be accepted.
- Collaboration is permitted, but you must write the solutions *by yourself without assistance*, and be ready to explain them orally to a member of the course staff if asked. You must also identify your collaborators. *Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.*
- To facilitate grading, please write down your solution to each problem on a separate sheet of paper. Make sure to include all identifying information and your collaborators on each sheet. Your solutions to different problems will be graded separately, possibly by different people, and returned to you independently of each other.
- 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.

Exercises These should not be handed in, but the material they cover may appear on exams:

1. Chapter 7, exercises 1–15 (in particular: 1–4, 8, 12.
2. Give a reduction from Weighted Interval Scheduling (Chapter 6) to the problem of finding the longest path in a DAG.
3. Give a reduction from the transactions matching problem of Chapter 4, Problem 16 to maximum bipartite matching.

Problems to be handed in

Page limits: The answer to each problem should fit in **2 pages (or one double-sided sheet)** of paper. Longer answers will be penalized.

1. KT, Chapter 7, Problem 5.
2. KT, Chapter 7, Problem 7.
3. We have seen two variants of bipartite "matching" problems so far: the stable marriage problem (Chapter 1), and the maximum bipartite matching (MBM) problem (Chapter 7.5). Your friend Greedy Gary thinks that the formalism of flows and cuts is overkill for understanding and solving the MBM problem. Instead, Gary suggests the following *reduction* from MBM to stable marriage:

Given a bipartite, unweighted graph G, with n nodes on the left and n nodes on the right, create an instance of the stable marriage problem (call it S) where men correspond to left nodes in G, and women correspond to right nodes in G. For convenience, let m_u be the node corresponding to (left) vertex u, and w_v be the woman corresponding to (right) vertex v; similarly, let u_m and v_w denote the vertices corresponding to m and w, respectively. The preference lists for S are created as follows: every man m rates acceptable matches (that is, women w such that (u_m, v_w) is an edge in G) above non-acceptable matches (that is, women for which the corresponding nodes are not connected); ties are broken arbitrarily. Similarly, women rate men to whom they are connected in G above men to whom they are not connected, breaking ties arbitrarily.

Gary claims that if we find a stable matching π for the instance S, then we can obtain a maximum matching for G by taking all edges in G for which the corresponding people are paired up in the stable matching (i.e. add an edge $e = (u, v)$ to the output if and only if m_u is paired with w_v in π).

(a) Suppose that Gary's claim is true. What is the running time of his new MBM algorithm? For which types of graphs is it better than the Ford-Fulkerson algorithm?

(b) Give either a proof of, or a counter-example to, Gary's claim.