Faculty of Science

FINAL EXAMINATION

COMPUTER SCIENCE COMP 360

Algorithm Design Techniques

Leacock 210 Monday December 10, 2007, 2-5pm

Examiner: Prof. D. Avis

Associate Examiner: Prof. A. Vetta

Instructions

Calculators are not allowed

No cell phones, books, or notes.

This exam consists of 6 questions, and has 3 pages including this cover page.

Dictionaries are not allowed

Answer in EXAM BOOK

1. (10 pts)

(a) Describe the stable marriage problem, and define what is meant by a *stable matching*

(b) Describe briefly an efficient algorithm for finding a stable matching.

(c) Consider a problem with 3 women A,B,C and 3 men a,b,c. The preference lists are:

$$A: abc \quad B: bca \quad C: acb$$

$$a: ABC \quad b: CBA \quad c: BAC$$

This means, for example, that A prefers a to b to c. Find two **distinct** stable matchings.

2. (10 pts) A popular psychiatrist has n clients for a given day. For $i = 1, \ldots, n$, client i has a requested start time s_i, finish time f_i and is willing to pay a consultation fee of w_i for the period. Design a **polynomial time** algorithm that will allow the psychiatrist to choose the subset of clients that maximizes her total fee on that day. Derive the time complexity of your algorithm.

3. (10 pts)

(a) Suppose you are given a directed graph with edge capacities, and a feasible flow between two given vertices s and t. Describe an algorithm that runs in linear time (in the input size) and verifies if the flow is the maximum flow between s and t. It should either give a certificate of optimality, or a flow that is larger than the given flow.

(b) It is required to find an invigilator for each of a consecutive set of m exams, labelled $e_1, \cdots, e_m$. There are n invigilators available, and for each invigilator there is a subset S_i of the exams that he/she could invigilate. No invigilator should invigilate more than three different exams. Show how to efficiently solve this problem by formulating it as a network flow problem. If for a given input there is no feasible assignment, what certificate of infeasibily could be given?

4. (10 pts)

(a) Explain what is a linear program in standard form, and what is its dual. State the strong duality theorem.

(b) Write down the dual of the following LP.

$$\text{maximize} \quad z = 5x_1 + 15x_2$$

$$s.t. \quad 2x_1 - x_2 \le 2$$

$$3x_1 + x_2 \le 5$$

$$-2x_1 + x_2 \le 4$$

$$x_1 \ge 0, \quad x_2 \ge 0$$

(c) It has been claimed that $x_1 = 0, x_2 = 4$ is a primal optimal solution, and that $y_1 = 0, y_2 = 7, y_3 = 8$ is a dual optimal solution. Either prove this claim, or if it is false, use the information given to obtain good bounds on the optimum objective function value.

5. (10 points)
3-SCHED: You are given the processing times $t_1, t_2, \cdots, t_n$ of n jobs, and a deadline T. All data are integers. Each of the jobs must be scheduled on one of three identical machines. Each machine can process one job at a time. Is there a schedule so that all jobs finish before time T?

(a) Define the class NP, and determine whether 3-SCHED is in NP.

(b) Explain what is meant by a polynomial time reduction.

(c) Reduce one of the following problems to 3-SCHED.
3-SAT, PARTITION, VERTEX COVER.
Give a description of the problem from the list that you have chosen, give the reduction, and prove it is correct.

(d) Describe an efficient approximation algorithm for the optimization version of 3-SCHED, i.e., find the smallest T such that all jobs are finished by time T. Your algorithm should obtain a bound on ratio of the approximate solution to the optimum solution that is at most 5/3. Give an example that illustrates the worst case behaviour of your algorithm.

6. (10 points)

In the max cut problem we are given an undirected graph $G = (V, E)$ and a positive weight $w(e)$ for each edge $e \in E$. For any partition the vertex set into subsets A and B we define the weight of the cut (A, B) to be the sum of the weights of the edges with one endpoint in A and one in B. The goal is to find the maximum weight cut.

Give either a local search algorithm or a greedy algorithm that finds a cut (A, B) which has weight at least half of the weight of the maximum weight cut. Prove that your algorithm delivers this performance guarantee.