Department of Mathematics
II Semester 2009-2010
MAL 342 Analysis and Design of Algorithms
Major Test Weightage 45%
Date 4.5.10 Time 1 P.M. – 3 P.M.

Q1. Let M be a matching in G=(V,E). Define M-augmenting path. Prove that a matching M in a graph G=(V,E) is a maximum matching if and only if there is no M-augmenting path in G. [1+2+3]

Q2. Define a flow in a Network. Describe how the maximum matching problem in bipartite graph can be solved using an algorithm for Maximum flow. [2+3]

Q3. Prove that chromatic number decision problem is NP-complete. [6]

Q4. Design a 2-approximation algorithm for TSP in which the weight function satisfies the triangle inequality. [6]

Q5. Design a divide and conquer based algorithm to construct a binary tree given the preorder numbering P[1..n] and inorder numbering I[1..n] of the binary tree. Prove the correctness of your algorithm and analyze your algorithm. [5]

Q6 A unit-time task is a job, such as a program to be run on a computer, that requires exactly one unit of time to complete. Given a finite set S of unit-time tasks, a Schedule for S is a permutation of S specifying the order in which these tasks are to be performed. The first task in the schedule begins at time 0 and finishes at time 1, the second task begins at time 1 and finishes at time 2, and so on.

The problem of scheduling unit-time tasks with deadlines and penalties for a single processor has the following inputs:

1. A set $S=\{a_1,a_2,\ldots,a_n\}$ of n unit time tasks;
2. A set of n integer deadlines $d_1,d_2,\ldots,d_n$, such that each d_i satisfies $1 \leq i \leq n$ and task a_i is
   supposed to finish by time d_i; and
3. A set of n nonnegative weights or penalties $w_1,w_2,\ldots w_n$, such that we incur a penalty of w_i if
   task a_i is not finished by time d_i and we incur no penalty if a task finishes by its deadline.

The unit-time task scheduling problem is to find a schedule that minimizes penalty incurred for missed deadlines.

Model this problem as a maximum weight independent set problem in some appropriate weighted Matriod [6]

Q7. Design a dynamic programming based algorithm to find the longest path in a Directed acyclic graph. Provide the proof of correctness and find the complexity of your algorithm if the DAG is given in adjacency list representation. [6]

Q8. Let S be a set of n elements. For any $x \in S$, the rank of x, r(x), is $|\{y \in S \mid y \leq x\}|$. Given S in the form of an array A and two numbers m_1 and m_2, find S_1, where $S_1=\{y \mid m_1 \leq r(y) \leq m_2\}$ in O (n) time. [5]
