CSE 551 Foundations of Algorithms
Final, Fall 2016
Closed Books, Closed Notes
Time: 1 Hour 50 minutes
Answer any five questions
Each question carries 20 pts.

Problem 1: Suppose there is a set A of men and a set B of women. Each set contain n elements. There exist two $n \times n$ arrays P and Q such that $P(i, j)$ is the preference of man i for woman j and Q is the preference of woman i for man j . Is it possible to find a a pairing of men and women in such a way that in the pairing there doesn't exist any man-woman (m-w) pair such that man m and woman w prefer each other to their assigned partners. If your answer is "yes" provide an algorithm to find such a pairing and also prove that your algorithm guarantees that no (m-w) pair will be in the computed pairing where man m and woman w prefer each other to assigned partners. If your answer is "no" provide an instance where such a pairing is impossible to find.

Problem 2: The n-th Fibonacci number is defined by the recurrence relation $F(n) = F(n-1)+F(n-2)$, with initial conditions $F(0) = 0$ and $F(1) = 1$. $F(n)$ can easily be computed with an algorithm with complexity $\theta(n)$. Develop an algorithm to compute $F(n)$ with complexity $\theta(log\ n)$. Explain the idea of your algorithm and analyze the algorithm to show that the complexity is indeed $\theta(log\ n)$. Show all your work.

Problem 3: Using Dynamic Programming technique, construct an optimal binary search tree for the following set of keys: C G K Q. The frequencies of successful (b_i's) and unsuccessful (a_i's) searches are given below:

$a_0 = 0, a_1 = 34, a_2 = 40, a_3 = 22, a_4 = 31$

$b_1 = 39, b_2 = 34, b_3 = 27, b_4 = 23$

Show all your work.

Problem 4: Suppose that we need to develop an algorithm for the product of two $n \times n$ matrices, where n is a power of 3. Using divide and conquer technique, the problem can be reduced to the multiplication of 3×3 matrices. The conventional method to do this requires 27 multiplications.
(i) With how many multiplications should one be able to do this so that the overall complexity of the algorithm is $O(n^{2.81})$?
(ii) With how many multiplications should one be able to do this so that the overall complexity of the algorithm is $O(n^{2.81})$ for the case where n is a power of 4 instead of 3 as in (i) ?

Problem 5: One of the basic motivations behind the Minimum Spanning Tree Problem is the goal of designing a spanning network for a set of nodes with minimum *total* cost. Here we explore another type of objective: designing a spanning network for which the *most expensive* edge is as cheap as possible.

Specifically, let $G(V, E)$ be a connected graph with n vertices, m edges, and positive edge costs that you may assume are all distinct. Let $T(V, E')$ be a spanning tree of G; we define the *bottleneck edge* of T to be the edge of T with the greatest cost.

A spanning tree T of G is a *minimum-bottleneck spanning tree* if there is no spanning tree T' of G with a cheaper bottleneck edge.
(a) Is every minimum-bottleneck tree of G a minimum spanning tree of G? Prove or give a counter-example.
(b) Is every minimum spanning tree of G a minimum-bottleneck tree of G? Prove or give a counter-example.

Problem 6: (i) Define the classes, P, NP and NP-complete. Show all your work.

(ii) 5-SAT is a version of the SATISFIABILITY problem, where all the clauses are restricted to have exactly five variables. Prove that 5-SAT is NP-complete.
(iii) Prove that the problem of determining if a graph $G = (V, E)$ has a Clique of size K, is NP-complete. Show all your work.