Major Exam (COL 702)

Read the instructions carefully:

- You need to justify correctness and running time of each algorithm.
- If using dynamic programming, you must explain the meaning of table entries, and explain the order of computing them.
- No argument should use examples – they will be ignored.
- You can assume any result proved during the lectures (but cannot assume any other result which is in the book or elsewhere).

1. For each of these problems, state whether they are true or false, and give a justification for your answer. There is no negative marking, but you will get marks only if the justification is correct.

   (i) (**2 marks**) If a decision problem in $\mathcal{P}$ is NP-complete, then $\mathcal{P} = NP$. Recall that $\mathcal{P}$ denotes the decision problems which have polynomial time algorithms.

   (ii) (**2 marks**) Let G be a connected undirected graph. We say that a vertex x is a safe vertex in G if removing x does not disconnect G. A vertex x is a safe in G if and only if the DFS tree formed by running DFS from x results in a DFS tree such that the root (i.e., x) has only one child.

   (iii) (**2 marks**) If $f(n) = O(g(n))$ then $2^{f(n)}$ is $2^{O(g(n))}$.

2. (**5 marks**) You are given a set S of $n+1$ distinct numbers (which may not be integers). You can assume that S is given as an array (need not be sorted). You are also given an unsorted array A of size n containing exactly n out of the $n+1$ numbers in S. Give an $O(n)$ time divide and conquer algorithm to find the number from S which is not in A. The only operation allowed on numbers in S (or A) is comparison (you are NOT allowed to perform addition, subtraction, multiplication, etc. on these numbers).

3. A shuffle of two strings X and Y is formed by interspersing the characters into a new string, keeping the characters of X and Y in the same order. For example, the strings PRODGYRNAMAMMIINCG and DYPRONGARMAMMICING are both shuffles of DYNAMIC and PROGRAMMING.

   (a) (**5 marks**) Given 3 strings $A[1..n], B[1..m], C[1..(n+m)]$, give a dynamic programming algorithm to determine whether C is a shuffle of A and B (the algorithm outputs a boolean value only).

   (b) (**4 marks**) A smooth shuffle of X and Y is a shuffle of X and Y that never uses more than two consecutive symbols of either string. For example, PRDOYGNARAMMMIICNG is a smooth shuffle of the strings DYNAMIC and PROGRAMMING, but DYPRONGARMAMMICING is not. Describe and analyze an algorithm to decide, given three strings $X[1..n], Y[1..m], Z[1..(n+m)]$, whether Z is a smooth shuffle of X and Y.

4. (**5 marks**) A town has f families, and k clubs. The i^{th} club needs a_i members, but can take at most 3 members from any family. Further, each person can belong to at most 1 club. Assume that the j^{th} family has b_j members. Show how the problem of assigning people to clubs while satisfying the above constraints can be expressed as a maximum flow problem.

5. (**5 marks**) We say that a set S of vertices in an undirected graph G forms a **near-clique** if there are edges between every pair of vertices in S, except perhaps for one pair – so a near-clique on k vertices will have either $\binom{k}{2}$ edges (in which case, it will be a clique) or $(\binom{k}{2} - 1)$ edges. Given a graph G and parameter $k > 0$, we would like to decide if the graph has a near-clique of size k – call this the NEAR-CLIQUE problem. Prove that the CLIQUE problem is polynomial time reducible to the NEAR-CLIQUE problem (recall that in the CLIQUE problem, we are given a graph G and a parameter k, and would like to decide if G has a clique of size k).

6. (**5 marks**) You are given two sets X and Y of n positive integers each. You are asked to arrange the elements in each of the sets X and Y in some order. Let x_i be the i^{th} element of X in this order, and define y_i similarly. Your goal is to arrange them such that $\Pi_{i=1}^{n} x_i^{y_i} = x_1^{y_1} \times x_2^{y_2} \times \cdots \times x_n^{y_n}$ is maximized. Give an efficient greedy algorithm to solve this problem.