CS 6212 – Mid Term

***Name:***

***Score: / 60***

(60 points) (120 minutes) (No calculators allowed) **[5 BONUS POINTS.]**

Q 1 – 6 (5 points each): ***[Only write answers, NO explanations]***

What is the time complexity of these algorithms, in terms of n?
***[ Sum += y is a short form notation for Sum = Sum + y.]***

|  |  |  |
| --- | --- | --- |
| for (int i = 1 to n) { for (int j = i to n) { for (int k = j to n) { Sum += a[i]\*b[j]\*c[k] } If (gcd(i,j) == 1) { j = n } }} | j = 1while (j < n) { k = j while (k < n) { Sum += a[k]\*b[k] k += k } j += 0.25 \* n} | int j = 2while (j < n) { int k = j while (k < n) { Sum += a[k]\*b[k] k += n1/3 log n } j = j\*sqrt(5)} |
|  |  |  |

Solve recurrence relation: T(n) = T(n/2) + O(n log n).

Solve recurrence relation: T(n) = 3T(n/2) + O(log n).

Give an example of a graph that has all 3 of the following properties. (Note that you need to give a single graph as the answer.)

1. It has one articulation point.
2. The graph needs at least 4 colors for a valid vertex coloring
3. The graph does not have a 4-clique (that is, a clique of 4 vertices) as a subgraph.

Q7 (10 points): Given two DNA sequences (sequences of A, C, T, G characters), the longest common subsequence (LCS) problem is to find the longest subsequence (not necessarily contiguous) that exists in both of the input strings. For example, given strings “ACTGACT” and “CAAGCATA”, the subsequence “CGCT” is common in both. Given two DNA sequences of sizes *n1* and *n2* respectively, find a dynamic programming algorithm to find the longest common subsequence in *O(n1n2)* time.

Q8 (10 points): ***“Rectangles”*** A rectangle can be specified in a 2-dimensional plane using the top left (north west) point and the bottom right (south east) point. Given n rectangles (using 2 points each), give an O(n log n) algorithm that tells if any two rectangles from the list overlap. Two rectangles are said to overlap, if there is a common point in both of them. [If a rectangle is entirely contained in another, they are still said to “overlap”, even though their lines don’t cross.]

Q9 (10 points): ***Long chain of friends:*** You are given a list of people, and statements of the form “x knows y”. You are asked to find, is there a chain of k people, such as x1 knows x2, x2 knows x3, and xk-1 knows xk. Prove that this problem is NP-complete by using one of the known NP-complete problems (CLIQUE, 3-SAT, Hamiltonian Path, Hamiltonian Cycle, Independent Set, etc.)