MTL 342, Design and Analysis of Algorithms, Assignment 1

The points will be for correctness and clarity of arguments as well as the readability of the write-up. Unnecessarily long answers might be penalized with negative points. For an algorithm, a proof of correctness is necessary to get the points.

Question 1.1
Here are two different definitions for the big-Oh notation. Assume that the functions f and g are non-negative and are defined for every natural number.

Definition 1. $f(n) = \mathcal{O}(g(n))$ if there exists a constant C such that for all $n \in \mathbb{N}$, the inequality $f(n) \leq C \cdot g(n)$ holds.

Definition 2. $f(n) = \mathcal{O}(g(n))$ if there exist constants n_0 and C such that for all $n \geq n_0$, the inequality $f(n) \leq C \cdot g(n)$ holds.

(i) Show that the above two definitions are equivalent.

(ii) For each of the following, decide whether they are true or not. Give reasons to support your answer.

(a) $n^2 = \mathcal{O}(n^2 \ln n)$

(b) If $f(n) = \mathcal{O}(g(n))$ then $2^{f(n)} = O(2^{g(n)})$.

(c) $\sum_{i=1}^{n} i^8 = \Theta(n^9)$

(d) $\sum_{i=1}^{n} \sqrt{i} = \Theta(n^{3/2})$

(iii) (Optional) Find two nondecreasing functions f and g defined for all natural numbers such that neither $f(n) = \mathcal{O}(g(n))$, nor $g(n) = \mathcal{O}(f(n))$.

[3 points]

Question 1.2
The distance between a pair of vertices u and v in an unweighted graph is the length (number of edges) of the shortest path between u and v. The diameter of a graph is the maximum distance between any pair of vertices. Give a linear-time algorithm to compute the diameter of a tree.

[3 points]

Question 1.3
Suppose that an n-vertex undirected unweighted graph $G = (V, E)$ contains two nodes s and t such that the distance between s and t is strictly greater than $n/2$. Show that there must exist some node v, not equal to either s or t, such that deleting v from G destroys all paths between s and t. Give an algorithm with running time $\mathcal{O}(m + n)$ to find such a node v. [3 points]

Question 1.4
Consider the following two change-making problems.

a) The input to this problem is an integer L. The output should be the minimum cardinality collection of coins required to make L paisa of change (that is, you want to use as few coins as possible). The coins are worth 1, 5, 10, 20, 25, and 50 paisa. Assume that you have an unlimited number of coins of each type. Formally prove or disprove that the greedy algorithm (that takes as many coins as possible from the highest denominations) correctly solves the Change Problem. So for example, to make a change for 234 paisa the greedy algorithms would take four 50 paisa coins, one 25 paisa coin, one 5 paisa coin, and four 1 paisa coins.

b) The input to this problem is an integer L. The output should be the minimum cardinality collection of coins required to make L nibbles of change (that is, you want to use as few coins as possible). Now the coins are worth 1, 2, 4,, 8, ..., 2^{1000} nibbles. Assume that you have an unlimited number of coins of each type. Prove or disprove that the greedy algorithm (that takes as many coins of the highest value as possible) solves the Change Problem.

[3 points]

Question 1.5
Consider the following problem. The input consists of the lengths $l_1, \ldots, l_n$ and access probabilities $p_1, \ldots, p_n$ for n files $F_1, \ldots, F_n$. The problem is to order these files on a tape so as to minimize the expected access time. If the files are placed in the order $F_{s(1)}, \ldots, F_{s(n)}$, then the expected access time is $\sum_{i=1}^{n} \left(p_{s(i)} \cdot \sum_{j=1}^{i} l_{s(j)}\right)$. Note that the term $p_{s(i)} \cdot \sum_{j=1}^{i} l_{s(j)})$ is the probability that you access the ith file times the length of the first i files. For each of the below algorithms, either give a proof that the algorithm is correct using an exchange argument, or a proof that the algorithm is incorrect.

a) Order the files from shortest to longest on the tape. That is, $l_i < l_j$ implies that $s(i) < s(j)$.

b) Order the files from most likely to be accessed to least likely to be accessed. That is, $p_i < p_j$ implies that $s(i) > s(j)$.

c) Order the files from the smallest ratio of the length over access probability to the largest ratio of the length over access probability. That is, $l_i/p_i < l_j/p_j$ implies that $s(i) < s(j)$.

[5 points]

Question 1.6
Consider the following problem. The input is a collection $A = a_1, \ldots, a_n$ of n points on the real line. The problem is to find a minimum cardinality collection S of unit intervals that cover every point in A. For each of the below two strategies, either show that they always produce an optimal solution, or prove that they do not do so.

a) Let I be the interval that covers the most number of points in A. Add I to the solution set S. Then recursively continue on the points in A not covered by I.

b) Let a_j be the smallest (leftmost) point in A. Add the interval $I = (a_j, a_{j+1})$ to the solution set S. Then recursively continue on the points in A not covered by I.

[3 points]

Question 1.7
(Optional) For the problem "Deep Down Below" described here, design an algorithm that finds the smallest possible power the hero must start the level with to be able to enter all the caves in some order and beat all the monsters. (Thanks to Mani Tyagi for introducing the problem).