Theory Qual

Please answer 3 of the 6 problems given. If you worked on more than 3 problems, make sure which 3 you would like us to grade. Only 3 problems will be graded.

1. A linear programming problem is specified by the following data:

$$\begin{array}{rrcl} & A\mathbf{x} & \leq & \mathbf{b} \\ & \mathbf{x} & \geq & 0 \\ \text{Maximize} & f & = & \mathbf{c}^T\mathbf{x} \end{array} \tag{1}$$

(where A is an $m \times n$ matrix, $\mathbf{b}$ and $\mathbf{c}$ are m- and n-dimensional vectors respectively, and $\mathbf{x}$ is an n-dimensional vector. T stands for transpose.) Similarly a linear programming problem can be in the form of minimizing a linear function, subject to a set of linear constraints.

The problem (1) has a (minimizing) linear programming problem called its dual program:

$$\begin{array}{rrcl} & A^T\mathbf{y} & \geq & \mathbf{c} \\ & \mathbf{y} & \geq & 0 \\ \text{Minimize} & g & = & \mathbf{b}^T\mathbf{y} \end{array} \tag{2}$$

The Duality Principle says that $\max f(x) = \min g(y)$.

Suppose we have just found a super fast probabilistic algorithm $\mathcal{A}$ which, for every instance of the linear programming problem, finds an optimal solution with probability at least $1/2$. (However, with probability up to $1/2$, it might produce a feasible but suboptimal solution with no warning.)

Find a way to rectify this probabilistic error, i.e., find an algorithm $\mathcal{B}$ which never produces a suboptimal solution. Your algorithm $\mathcal{B}$ can be probabilistic and should produce an optimal solution with probability at least $1 - 1/2^{100}$. (With probability up to $1/2^{100}$, your algorithm $\mathcal{B}$ can terminate without a definite solution.) Estimate the running time of your algorithm $\mathcal{B}$ in terms of the running time of algorithm $\mathcal{A}$.

2. It is known that the following subset sum problem is NP-complete:

Instance: A sequence of positive integers $a_1, a_2, \ldots, a_n$, and a positive integer b.
Problem: Are there $x_1, x_2, \ldots, x_n \in \{0, 1\}$, such that

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b?$$

It is also known that the problem remains NP-complete if we ask for more "general" solutions $x_1, x_2, \ldots, x_n \in \{0, 1, 2\}$ in the above equation.

Now let's suppose we ask for the most "general" version of the problem where solutions are restricted to be integers:
(*)Problem: Are there $x_1, x_2, \ldots, x_n \in \mathbf{Z}$, such that

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b?$$

(A) What is the fallacy in the following statement:

Since this last version is asking for a more general solution, the problem is also NP-complete.

(B) Is the problem (*) NP-complete? If so, give a proof. If not, give a polynomial-time algorithm.

3. The 0-1 Knapsack Problem can be defined as follows. A bank robber has entered a bank vault and finds n items; the ith item is worth v_i dollars and weighs w_i pounds. He wants to take as valuable a load as possible, but can only carry W pounds in his knapsack. Which items should he take? [Assume all the numbers above are integers.]

   (A) Describe a dynamic programming solution for the 0-1 knapsack problem.

   (B) Most dynamic programming algorithms run in polynomial time. What is the complexity of your algorithm?

   (C) Since the 0-1 knapsack problem is NP-complete, why doesn't your algorithm show that P = NP?

4. You are at the origin of a 2-dimensional grid. A treasure is located at some grid point that is distance n from you but you do not know where it is. Your goal is to find it. To do this you may repeatedly move from your current location to one of its 4 neighbors, to the left, to the right, above, and below. Each such move costs one time unit. If you are at location (x, y) you can tell whether the treasure is located there.

   (A) a) Give a strategy that will find the treasure in time $O(n)$, assuming you know that only horizontal moves are necessary, where the treasure if eventually found to be located in location $(\pm n, 0)$.

   (B) For the general problem, give a strategy that uses time $O(n^2)$.

   (C) Prove that no randomized strategy using time $o(n^2)$ can find the treasure with $\geq 1\%$ probability.

5. Let $x_1, x_2, \ldots, x_n$ be binary strings of length $\leq m$, not necessarily distinct. Design and analyze an efficient procedure that counts the pairs (i, j) with $i < j$ and $x_i = x_j$. Your procedure should be as efficient as possible.

6. Prove: If NP is contained in BPP, then NP = RP.