CMPSCI 611: Advanced Algorithms
Micah Adler

Problem Set 5

Out: November 30, 2000
Due: December 7 2000, 2000

1. [CLR] Problem 37-1 (page 983). For part (a), you can assume that the following Partition problem is NP-Complete:
   INPUT: a set of positive integers $A = \{a_1 \ldots a_n\}$.
   QUESTION: can the set A be partitioned into two sets S and $A - S$ such that the sum of the integers in S is equal to the sum of the integers in $A - S$?

2. We say that there is a fully polynomial time approximation scheme (FPTAS) for a problem if there is an approximation algorithm that takes as input an instance of the problem and a value $\epsilon > 0$, and returns a solution that is within a factor of $1 + \epsilon$ of optimal. The running time of the algorithm must be polynomial in the size of the input, as well as in $\frac{1}{\epsilon}$.

   We say that a problem is NP-Complete in the strong sense if the problem remains NP-Complete even when we restrict all numbers appearing in the input to be polynomial in the size of the input.

   Suppose that a strongly NP-Complete maximization problem has the property that for all inputs x, the optimum cost is bounded by $p(NUM(x))$, where $p()$ is a polynomial, and $NUM(x)$ is the largest number appearing in the input x. Show that if there is a FPTAS for such a problem, then $P = NP$.

   You can assume that all numbers appearing in the input x are integers.

3. [CLR] Problem 37-2 (page 984). In addition, part (c) is to use what you proved in Question 2 to show that if there is a c-approximation algorithm for the max-clique problem, for any constant c, then $P = NP$.

   Note that there is a mistake in the way that the graph $G^{(k)}$ is defined in part a. In particular, $E^{(k)}$ should be defined so that $(v1, v2, \ldots, vk)$ is adjacent to $(w1, w2, \ldots, wk)$ if and only if FOR EACH i, $1 \leq i \leq k$, either vertex v_i is adjacent to v_i in G, or else $v_i = w_i$. (The book states that edges exist if this holds for SOME i.)

4. (a) Invent an example of a problem that can be expressed as a linear programming problem. Everyone should invent their own example, and not use one covered in class, or found from any other source.

   (b) Consider the following linear programming problem:

$$\begin{aligned} \text{maximize } 2x_1 - 3x_2 \\ \text{s.t. } x_1 - 5x_2 \leq 1 \\ -2x_1 + x_2 \leq 2 \\ x_1 - x_2 \leq 7 \\ x_1 - 2x_2 \leq 4 \\ x_1 - 3x_2 \leq 3 \end{aligned}$$

   Sketch the feasible polyhedron for this problem, showing clearly the facets and the vertices (with coordinates). Find the optimal solution to the problem, explaining your method.

   (c) Convert the problem of part (b) to standard form by adding surplus variables $x_3, \ldots, x_7$ to the inequalities.