CS 1510 Midterm 3

Fall 2003

1. (20 points) Consider the following 3Clique problem:

   INPUT: A undirected graph G and an integer k.

   OUTPUT: 1 if G has three vertex disjoint cliques of size k, and 0 otherwise.

   Show that this problem is NP-hard. Use the fact that the clique problem in NP-complete. The input to the clique problem is an undirected graph H and an integer j. The output should be 1 if H contains a clique of size j and 0 otherwise. Note that a clique is a mutually adjacent collection of vertices. Three cliques are disjoint if no vertex is in more than one clique.

2. (20 points) Show that the Vertex Cover Problem is self-reducible. The decision problem is to take a graph G and an integer k and decide if G has a vertex cover of size k or not. The optimization problem takes a graph G, and returns a smallest vertex cover in G. So you must show that if the decision problem has a polynomial time algorithm then the optimization problem also has a polynomial time algorithm. Recall that a vertex cover is a collection S of vertices with the property that every edge is incident to a vertex in S.

3. (20 points) Consider the following Bahncard problem where the input is a collection of n train trips within Germany. For the ith trip T_i you are given the date d_i of that trip, and the non-discounted fare f_i for that trip. For simplicity we will assume that dates are nonnegative integers. The German railway system sells a Bahncard for Y Euros that entitles you to a 50% fare reduction on all train travel within Germany within A days of purchase. You can apply at most one Bahncard discount to a particular trip. The problem is to determine the least you can spend on your travel.

   In this paragraph we give an example. Assume that $Y = 50$, $A = 100$. Further assume, $d_1 = 1$, $f_1 = 20$ Euros, $d_2 = 40$, $f_2 = 200$ Euros, $d_3 = 80$, $f_3 = 200$ Euros, $d_4 = 120$, $f_4 = 100$ Euros, $d_5 = 200$, $f_5 = 200$ Euros, and $d_6 = 400$, and $f_6 = 600$ Euros. Then you might buy a Bahncard on day 40, and day 400. This results in a total cost of

   $$20 + 50 + (.5)(200) + (.5)(200) + (.5)(100) + 200 + 50 + (.5)(600)$$

   Give an $O(n^2)$ time algorithm for problem. Significant partial credit will be given for any polynomial time algorithm.