CS466/666 Spring 2013
Assignment 6
(Due Noon, Monday June 24, 2013)

You are on your honour to present your own work and acknowledge your sources.

1. First we will denote a running time of $O(f(k)n^c)$ where c is a constant, by $O^*(f(k))$ (i.e. O^* notation ignores polynomial factors of n).
   In assignment 5, you were asked to give an $O^*(3^k)$ algorithm to determine whether a given tournament on n vertices has a feedback vertex set of size at most k. The purpose of this exercise is to improve this bound to $O^*(2^k)$ using the method of iterated compression.

   (a) Show that the problem can be solved by fewer than n applications of an algorithm for the following compression problem: [2 marks]

   Given a tournament T on n vertices, a feedback vertex set S of size $k+1$ to the tournament, determine whether T has a feedback vertex set of size at most k.

   (b) Now we move on to solve the compression problem. Show that the compression problem can be solved by $2^{k+1}-1$ applications of the following problem which we call *Disjoint (T, S, k)*: [2 marks]

   Given a tournament T on n vertices, a feedback vertex set S of size at most $k+1$, determine whether T has a feedback vertex set disjoint from S and is of size less than $|S|$.

   (c) Now give a polynomial time algorithm for *Disjoint (T, S, k)* using the following hints. [5 marks]

   - We can assume that the 'subtournament' on the vertices of S and the subtournament on the vertices of $V(T)-S$ are acyclic. Why? (here $V(T)$ is the vertex set of the tournament T). Hence perform a topologic sort of each of the subtournaments (refer CLRS for topological sort).
   - What can you do if there is a triangle in T with exactly one vertex from $V(T)-S$?

- Now, for every vertex v in $V(T) - S$, find a natural position $p(v)$ with respect to the vertices in S. Sort the vertices in $V(T) - S$ using the $p(v)$ values. Let this order of vertices in $V(T) - S$ be π_1 and let the topological sorted order of vertices in $V(T) - S$ be π. Argue that we can now find the required minimum feedback vertex set (with all vertices from $V(T) - S$) by finding the longest common subsequence between π and π_1.

(d) Give the complete algorithm and argue its running time to be $O^*(2^k)$. Assume that topological sort of a list and longest common subsequence of two lists can be found in polynomial time. [1 mark]

2. The purpose of this exercise is to give a $O(k^3)$ kernel for the 3-hitting set problem. We are given a universe U on n elements, and m subsets of U each with 3 elements. The question is to dermine whether there is a subset of k elements of the universe that has a non-empty intersection with each of the m subsets (i.e. whether there are k elements that will 'hit' each of the m sets!).

   Show that in polynomial time either we can deduce that there is no such hitting set or we can reduce the given instance to an equivalent instance with $O(k^3)$ elements and sets. [5 marks]

   [The following hints can help you come up with preprocessing rules.

   - What can you say if a pair of elements (x, y) appear together in more than k sets?
   - After implementing a rule based on the answer to the above question, what can you say if an element appears in more than k^2 sets?]