Homework 6
Computer Science 511
Fall 2009
Due by 5 PM on Friday, December 4

Reading Assignment

Chapter 10 of K & T.

Problem Set

1. (10 points) Recall that the input to the ***set cover problem*** (SC) consists of (i) a pair $X = (U, \mathcal{F})$, where U is a set of n elements, and $\mathcal{F}$ is a collection of m subsets of U and (ii) a nonnegative integer k. The question is whether there exists a set $\mathcal{F}' \subseteq \mathcal{F}$ such that $|\mathcal{F}'| \leq k$ and $\bigcup_{S \in \mathcal{F}'} S = U$. (The set $\mathcal{F}'$ is called a ***set cover*** for X.)

   Here we consider ***3-SC***, the special case of SC where each element in U appears in at most 3 different sets in $\mathcal{F}$. Note that this problem is NP-complete, since it is in NP and includes Vertex Cover as a special case. We use the following notation. Let $\langle X = (U, \mathcal{F}), k \rangle$ be an instance of 3-SC.

   - For any $S \in \mathcal{F}$, $X - S$ denotes the instance of 3-SC obtained by removing S from $\mathcal{F}$, replacing U by $U - S$, and replacing S' by $S' - S$ for every $S' \in \mathcal{F} - \{S\}$.
   - For every element $v \in U$, $\mathcal{F}_v$ denotes the collection of sets in $\mathcal{F}$ that contain v. Note that $|\mathcal{F}_v| \leq 3$.

   (a) Let v be any element of U. Show that X has a set cover of size at most k if and only if there is some $S \in \mathcal{F}_v$ such that $X - S$ has a set cover of size at most $k - 1$.

   (b) Use the result of part (a) to give an algorithm for 3-SC with running time $O(c^k p(n, m))$, where c is a constant and $p(\cdot, \cdot)$ is a polynomial function.

2. (10 points) A ***perfect elimination ordering*** in a graph is an ordering of the vertices of the graph such that, for each vertex v, v and the neighbors of v that occur later than v in the order form a clique.

(a) Show that every tree has a perfect elimination ordering.

(b) Show that every triangulated cycle has a perfect elimination ordering.

(c) Suppose G is a graph of tree-width k, for some fixed k. Show that G may not have a perfect elimination ordering.

(d) Suppose $G = (V, E)$ has a tree decomposition $(T, \{V_t : t \in T\})$. The ***fill-in*** of G is the graph $G' = (V, E')$ obtained by adding all missing edges to V_t for every $t \in T$. That is, for every $t \in T$ and every $u, v \in V_t$, $(u, v) \in E'$. Show that the fill-in of G has a perfect elimination ordering.

Note. Perfect elimination orderings can be used to obtain linear-time algorithms on graphs of bounded tree-width. However, we will not pursue this idea here.

3. (10 points) Exercise 3, page 596.

4. (10 points) Exercise 6, page 597.

5. (10 points) Exercise 9, pages 597–598.

6. (10 points) As mentioned in class, a graph is ***planar*** if it can be drawn on the plane in such a way that no two of its edges intersect, except at their endpoints. It can be shown that an n-vertex planar graph has $O(n)$ edges. Much more remarkable is the following result proved by Lipton and Tarjan.

   The Planar Separator Theorem. *Let G be any n-vertex planar graph with non-negative vertex costs summing to no more than one. Then the vertices of G can be partitioned into three sets A, B, C such that no edge joins a vertex in A with a vertex in B, neither A nor B has total vertex cost exceeding $2/3$, and C contains no more than $2\sqrt{2n}$ vertices. Furthermore A, B, C can be found in $O(n)$ time.*

   Use the Planar Separator Theorem to obtain a $O(2^{c\sqrt{n}}\text{poly}(n))$ algorithm to solve the maximum independent set problem on planar graphs. Here c is a constant and poly$(\cdot)$ stands for some polynomial function. Note that, while the algorithm is exponential, its running time is a substantial improvement on ordinary exhaustive enumeration, which takes $O(2^n\text{poly}(n))$ time. Note also that maximum independent set remains NP-complete for planar graphs.

   Hint: Use the Planar Separator Theorem to show that an n-node planar graph has a tree-decomposition of width $O(\sqrt{n})$ and that this decomposition can be built efficiently. Then, apply the ideas in Section 10.4.