COMP3804 Winter 2006 — Assignment 4 — Due Thursday April 6

This assignment is due Thursday April 6th. Note, there is no class on that day, so your assignment should be placed in the COMP3804 assignment box in the CCSS lounge (4135 HP).

1 Applications of the Big Theorem

Here's a theorem from the big white book:

Theorem 1 (23.1) *Let $G = (V, E)$ be a connected, undirected weighted graph with a real-valued weight function w defined on E. Let A be a subset of E that is included in some minimum spanning tree of G, let $(S, V \setminus S)$ be any cut of G that respects A and let (w, v) be a light edge crossing $(S, V \setminus S)$. Then edge (u, v) is safe for A.*

Using the above theorem, prove the following statements about minimum spanning trees:

1. Let (u, v) be an edge of G with minimum weight. Then there exists a minimum spanning tree of G containing the edge (u, v).
2. Let (u, v) be an edge of G such that there is at most one edge of G whose weight is less than $w(u, v)$. Then there exists a minimum spanning tree of G containing the edge (u, v).

2 The Star Problem

A *star* is a person who doesn't know anyone in the room but everyone else in the room knows the star.

1. Prove that in any room containing n people there is at most one star.
2. Suppose you are only allowed to make the following query: You select two people A and B from the room and ask A "do you know B?" Show that, after asking $O(n)$ such questions you will be able to determine whether the room contains a star and who the star is.
3. Given the adjacency matrix[1] of a graph $G = (V, E)$ show how to find, in $O(|V|)$ time, if there is a vertex $|V|$ with $n - 1$ incoming edges and no outgoing edges.

[1] The adjacency matrix of a directed graph G is a $|V| \times |V|$ binary matrix B where $B_{i,j} = 1$ if and only the directed edge (i, j) is in G.

3 Reducing Clique to Independent-Set

The CLIQUE problem takes a graph $G = (V, E)$ and an integer k and asks if G contains a clique of size k. (A clique is a set of vertices such that every pair of vertices in the set is adjacent.)

The INDEPENDENT-SET problem takes a graph $G' = (V', E')$ and an integer k' and asks if G' contains an independent set of size k'. (An independent set is a set of vertices such that no pair of vertices in the set is adjacent.)

Give a polynomial time algorithm that, given a graph G and an integer k produces a graph G' and an integer k' such that G has a clique of size k if and only if G' has an independent set of size k'.

4 Reducing Vertex-Cover to Set-Cover

The VERTEX-COVER problem takes a graph G and an integer k and asks if G contains a vertex cover of size k. (A vertex cover of G is a subset of the vertices of G such that every edge of G is incident to some vertex in the subset.)

The SET-COVER problem takes a collection of sets $S_1, \ldots, S_n \subseteq \{1, \ldots, m\}$, an integer m and an integer k' and asks if there are k' sets $S_{i_1}, \ldots, S_{i_{k'}}$ such that $\cup_{j=1}^{k'} S_{i_j} = \{1, \ldots, m\}$.

Give a polynomial time algorithm that, given a graph G and an integer k produces a collection $S_1, \ldots, S_n$ of sets, an integer m and an integer k' such that there are k' sets $S_{i_1}, \ldots, S_{i_{k'}}$ such that $\cup_{j=1}^{k'} S_{i_j} = \{1, \ldots, m\}$ if and only if G has a vertex cover of size k.

5 Reducing Vertex-Cover to Dominating-Set

The DOMINATING-SET problem takes a graph G' and an integer k' and asks if G' contains a dominating set of size k'. (A dominating set is a subset of the vertices of G' such that every vertex of G' is in, or adjacent to, some vertex in the subset.)

Give a polynomial time algorithm that, given a graph G and an integer k produces a graph G' and an integer k' such that G has a vertex cover of size k if and only if G' has a dominating set of size k'.

6 Reducing Subset-Sum to Partition

The SUBSET-SUM problem takes a set $S = \{x_1, \ldots, x_n\}$ of positive integers and an integer t and asks if there is a subset $S' \subseteq S$ such that $\sum_{x \in S'} x = t$.

The PARTITION problem takes a list $X = \langle x_1, \ldots, x_n \rangle$ of positive integers and asks if there is a subset $X' \subseteq X$ such that $\sum_{x \in X'} x = \frac{1}{2} \sum_{x \in X} x$.

Give a polynomial time algorithm that takes as input (S, t) and outputs X such that the answer to the SUBSET-SUM instance (S, t) is yes if and only if the answer to the PARTITION instance X is yes.

7 Finishing the Job

Questions 3–6 asked you to show that Independent-Set, Set-Cover, Dominating-Set and Paritition are NP-hard. Finish the job of proving that these problems are NP-complete by showing that each of them is in NP.

8 Self-Reducibility of Vertex-Cover

Suppose we have an algorithm $\mathcal{A}$ for Vertex-Cover that runs in $O(T(n, m))$ time on graphs with n vertices and m edges.[2] More precisely, the algorithm takes a graph G and an integer k and outputs *true* if G has a vertex cover of size k and *false* otherwise. *The algorithm doesn't output anything else.*

1. Using algorithm $\mathcal{A}$ as a subroutine, describe an $O(T(n, m) \log n)$ time algorithm to determine the smallest value k^* such that G has a vertex cover of size k^*.

2. Notice that the above algorithms don't tell us the actual vertices in the vertex cover, just whether or not one exists. Show how, if we already know some value k for which G has a vertex cover of size k we can find the vertices of a vertex cover of size k in $O(nT(n, m))$ time.

[2] Assume $T(n, m)$ is a non-decreasing function of both n and m.