Complexity and Approximation Algorithms

Due: Friday, May 27th

Directions: Some of the questions on this assignment will appear on the quiz on Friday, May 27th.

1. State the following optimization problems as decision problems:
   - Vertex Cover
   - Independent Set
   - Knapsack
   - Longest Increasing Subsequence
   - Coin Row
2. Fill in the reduction diagram explicitly for the following reductions. You should include what a specific instance and solution looks like for each problem in the reduction.
   - Vertex Cover $\rightarrow$ Independent Set
   - Clique $\rightarrow$ Independent Set
   - 3SAT $\rightarrow$ Stingy SAT (see problem 6)
   - Vertex Cover $\rightarrow$ Set Cover (see problem 8)
   - Clique $\rightarrow$ Subgraph Isomorphism (see problem 7)
3. Consider the Clique problem restricted to graphs in which every vertex has degree at most 3. Prove that this problem is in NP.
4. Suppose that Vertex Cover is NP-Complete (it is), use that fact to show that Independent Set is NP-Complete.
5. Suppose that Clique is NP-Complete (it is), use that fact to show that Independent Set is NP-Complete.
6. Stingy SAT is the following problem: given a set of clauses (each a disjunction of literals) and an integer k, find a satisfying assignment in which at most k variables are true, if such an assignment exists.

   Prove that Stingy SAT is NP-complete. (Hint: Use 3SAT)
7. In the Subgraph Isomorphism problem we are given as input two undirected graphs G and H, we wish to determine whether G is a subgraph of H (that is, whether by deleting certain vertices and edges of H we obtain a graph that is, up to renaming of vertices, identical to G), and if so, return the corresponding mapping of $V(G)$ into $V(H)$.

   Prove that Subgraph Isomorphism is NP-complete. (Hint: Use Clique)
8. In the Set Cover problem, we are given a set ("universe") of elements: $\{1, 2, \ldots m\}$ and a set S of n sets whose union is the universe. The goal is to find a subset of S of size y whose union equals the universe. Note that this is the decision version of the Set Cover problem stated above.

   Prove that Set Cover is NP-complete. (Hint: Use Vertex Cover)
9. Design an approximation algorithm for the Set Cover Problem: Given a universe U of n elements, and a collection of subsets of U say $S = \{S_1, S_2, \ldots, S_m\}$ where every subset S_i has an associated cost, $c(S_i)$. Find a minimum cost subcollection of S that covers all elements of U. (Hint: Greedily choose the set whose cost effectiveness is best - the ratio of cost to newly added elements is minimum.)

10. Design a 2-approximation algorithm for the following problem: Given n cities and distances between every pair of cities (suppose these distances respect the triangle inequality), select k cities to place warehouses such that the maximum distance of a city to a warehouse is minimized. (Hint: Choose the first warehouse arbitrarily then greedily choose the warehouse that is farthest away from the current set of warehouses.)

11. Given an undirected graph $G = (V, E)$ in which each vertex has degree $\leq d$, show how to efficiently find an independent set whose size is at least $1/(d+1)$ times that of the largest independent set. (Hint: Consider the basic greedy algorithm of choosing the vertex of minimum degree.)

12. In the Minimum Steiner Tree problem, the input consists of:

    - A complete graph $G = (V, E)$
    - Distances d_{uv} between all pairs of vertices and
    - A distinguished set of terminal nodes $V' \subseteq V$

    The goal is to find a minimum cost tree that includes the vertices V'. This tree may or may not include vertices in $V \setminus V'$. Suppose the distances in the input are *metric*:

    - $d(x, y) \geq 0$ for all x and y.
    - $d(x, y) = 0$ if and only if $x = y$.
    - $d(x, y) = d(y, x)$.
    - $d(x, y) \leq d(x, z) + d(z, y)$

    Show that an efficient 2-approximation algorithm for Minimum Steiner Tree can be obtained by ignoring the nonterminal vertices and simply returning the minimum spanning tree on V'. (Hint: Recall our approximation algorithm for the TSP.)