Homework #1, CS 329E, Spring 2008
Solutions

Problems:

1. Phrase each of the following real world problems as a graph-theoretic problem:

   (a) You are given a collection of people, and you know which people know each other (which you assume is symmetric). You want to find a group of people who, between them, know everyone. You'd like this group to be as small as possible.

   Solution: Let the graph $G = (V, E)$ have vertex set V corresponding to the people in the collection, and let (v, w) be an edge in E if and only if the people denoted by v and w know each other. We want to find a set $A \subseteq V$ of minimum cardinality such that for all $v \in V - A$, $\exists a \in A$ s.t. $(v, a) \in E$. (Equivalently, we want to find a subset A of the vertices such that for all vertices v in V that are not in A, there is at least one node in A that is adjacent to v.)

   (b) It is 1900, and you are the match maker for a small village in the Ukraine. Thus, your job is to marry off all the unmarried young men and women. You know which pairs of men and women are willing to be married (and, more importantly, whether their parents are willing to let their children get married, which depends also upon the dowry and other factors). You get a fixed fee for each marriage you make. Under the assumptions that you cannot marry anyone to more than one person and you cannot perform any same sex marriages, you'd like to maximize the fee you'll be paid.

   Solution: Let the graph $G = (V, E)$ have vertex set V corresponding to each of the unmarried young men and women in the village, and let (v, w) be an edge if and only if v and w are willing to be married (and their parents are agreeable). We wish to find a maximum cardinality subset $E_0 \subseteq E$ such that for all pairs of edges $(x, y) \in E_0$ and $(x', y') \in E_0$, if $x = x'$ then $y = y'$. Equivalently, we want to find a maximum cardinality subset E_0 of the edges so that no two edges in E_0 share any vertices.

   (c) You are the social organizer for your friends' graduations, and you need to arrange a number of graduation parties. The only problem is that some people can't stand each other, and so you can't have them at the same party. The main expense in each party is renting the nightclub where the party takes place, and so you want to have as few parties as you can arrange.

   Solution: Let the graph $G = (V, E)$ be defined to have vertex set corresponding to your friends, and let E consist of those pairs (x, y)

of vertices corresponding to people who can stand each other. Here are three ways of expressing what we want:

- We want to find a proper vertex-coloring of the graph which uses the minimum number of colors (so that the vertex colors denote the parties).
- We want to partition the vertices into as few sets as possible so that no two vertices in the same set are adjacent (share an edge between them).
- We want to find a partition of the vertices into sets $A_1, A_2, \ldots, A_k$ with k as small as possible, so that for all i and for all $\{v, w\} \subseteq A_i$, the pair v, w is not connected by an edge (i.e., $(v, w) \notin E$).

2. Show how you'd use an oracle for the Yes/No problem to construct a solution to each of the construction problems below.

   (a) Given a graph, determine if it has a clique of size 5, and return it if it does.

   Solution: Begin by determining if the graph G has a 5-clique. If it does not, return "No". If it does, then do the following:

   - Enumerate the vertices of the graph, $v_1, v_2, \ldots, v_n$.
   - For $i = 1...n$ DO:
     - If $G - v_i$ has a 5-clique, then replace G by $G - v_i$.
   - Return V.

   (b) Given a set X of items of different weights, and given a specific target total weight B, determine if there is a subset of X of total weight B. (For example, if the input has set $X = \{1, 5, 8, 13, 21, 27, 30\}$ and $B = 41$, then the answer is *yes*, and the subset would be $\{1, 13, 27\}$.)

   Solution: Order the set X arbitrarily, $x_1, x_2, \ldots, x_n$. Ask if X has a subset of total weight B. If not, return "no". Else, do:

   - $A \leftarrow \emptyset$
   - For $i = 1...n - 1$ do:
     - If $x_{i+1}, x_{i+2}, \ldots, x_n$ has a subset of total weight $B - x_i$, then set $A \leftarrow A \cup \{x_i\}$ and set $B \leftarrow B - x_i$.
   - Return A.

   (c) Given a sequence of integers $a_1, a_2, \ldots, a_n$, find an increasing subsequence $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$ (so $i_1 < i_2 < \ldots i_k$) of maximum length k. (Thus, the solution to $1, 2, 5, 3, 4, 8, 5, 1, 9$ is $1, 2, 3, 4, 5, 9$, which has length 6.)

   Solution: First find the length of the longest increasing subsequence of the sequence $A = a_1, a_2, \ldots, a_n$, and set that to B. Then do the following:

   - For $i = 1...n - 1$ do:

- If $A' \leftarrow A - a_i$ has an increasing subsequence of length B then replace A by A'.

(d) (Extra Credit) Given a graph, show how to use an oracle for the decision problem (an answer to the question of whether it has a 3-coloring of the vertices) to construct the 3-coloring.

Solution: First determine if the graph G has a 3-coloring. If it does not, return nothing. Otherwise, do the following.

- Mark all pairs v, w of vertices which are not adjacent (i.e., $(v, w) \notin E$) as "untested."
- While G contains an untested pair of non-adjacent vertices v, w, Do:
  - Find one such pair v, w and *merge* them into a single vertex (which we will call v) which is adjacent to all vertices that were adjacent to either v or w. Ask if this graph has a 3-coloring. If it does not, then mark the pair v, w as "tested". If it does have a 3-coloring, then record that $c(v) = c(w)$.
- After all non-adjacent pairs of vertices have been tested, there will only be three vertices left in the graph, and we can use 3-color this graph. Each of these nodes in the final graph represents a subset of the original vertex set, and these are all given the same color.

3. Express each of the following functions recursively (i.e., give the value for $p(1)$ and define $p(n)$ in terms of $p(n-1)$):

   (a) $p(n) = n^3$.
   Solution: $p(1) = 1$ and $p(n) = p(n-1) + 3n^2 - 3n + 1$ if $n > 1$.

   (b) $p(n) = 2^n$.
   Solution: $p(1) = 2$ and $p(n) = 2p(n-1)$ if $n > 1$.

   (c) $p(n) = n!$
   Solution: $p(1) = 1$ and $p(n) = np(n-1)$ if $n > 1$.

   (d) $p(n) = n^2 + n$
   Solution: $p(1) = 2$ and $p(n) = p(n-1) + 2n$ if $n > 1$.

4. Pick any *one* of the functions from the previous problem, and prove that it equals its recursive definition, using induction.

   Solution: We prove this for $p(n) = 2^n$. Let $q(n)$ be defined by $q(1) = 2$ and $q(n) = 2q(n-1)$ if $n > 1$. Then we will prove that $p(n) = q(n)$ for all $n \geq 1$ by induction on n. The base case is where $n = 1$, and this we see by evaluating $p(1)$ and $q(1)$ and noting that both equal 2. Our inductive hypothesis (IH) is that $p(N) = q(N)$ for some specific N, and we use that to show that $p(N+1) = q(N+1)$. We examine $q(N+1)$. This is equal to $2q(N)$, by definition. By the inductive hypothesis, this is equal to $2p(N)$. By the definition of the function $p(n)$, $p(N) = 2^N$. Hence,

$q(N+1) = 2q(N) = 2p(N) = 2 * 2^N = 2^{N+1}$. But since $p(N+1) = 2^{N+1}$ (by definition), it follows that $q(N+1) = p(N+1)$, and our proof is complete.

5. For each of the following pairs of functions $f(n)$ and $g(n)$, state (without proof) which of the following cases is true: (a) $f(n)$ is $O(g(n))$, (b) $g(n)$ is $O(f(n))$, (c) each is big-oh of the other (i.e., $f(n)$ is $O(g(n))$ and $g(n)$ is $O(f(n))$), or (d) neither is big-oh of the other.

   (a) $f(n) = 3n^2, g(n) = 5n^3$.
   Solution: (a)

   (b) $f(n) = \log n, g(n) = 500$
   Solution: (b)

   (c) $f(n) = (\log n)^5, g(n) = n/2$,
   Solution: (a)

   (d) $f(n) = \sqrt{n}, g(n) = \log n$,
   Solution: (b)

   (e) $f(n) = \sqrt{n}, g(n) = n$,
   Solution: (a)

   (f) $f(n) =$
   - 1 if n is odd
   - n if n is even

   $g(n) = n$
   Solution: (a)

6. Prove that $3n^2$ is $O(n^2 + 100)$ (by producing the two constants).

   Solution: Let $C = 3$ and $C' = 1$. Then note that $3n^2 \leq C(n^2 + 100)$ for all $n \geq C'$, and we are done.