CS 3510 - Honors Algorithms
Homework 6
Assigned March 26
Due Tuesday, April 4

1. Generalize Huffman's algorithm to ternary codewords (i.e., codewords using the symbols 0, 1, and 2), and prove that it yields optimal ternary codes.

2. What is the optimal Huffman code for the following set of frequencies, based on the first 8 Fibonacci numbers?

$$a : 1, b : 1, c : 2, d : 3, e : 5, f : 8, g : 13, h : 21.$$

   Can you generalize your answer to find the optimal code when the frequencies are the first n Fibonacci numbers? Prove your answer.

3. Give verification algorithms to show that the following problems are in **NP**. Formulate each problem as a decision problem first, if necessary.

   (a) Longest Path: Given an undirected graph $G = (V, E)$ and nodes $u, v \in V$, what is the longest simple path between u and v?

   (b) Graph Coloring: Given an undirected graph $G = (V, E)$, what is the minimum number of colors for which one can assign a color to each vertex so that no two adjacent vertices have the same color.

4. Show that every problem in **NP** can be solved in exponential time. That is, in time $O(2^{p(n)})$ for some polynomial $p(n)$.

5. The MINIMUM-LEAF-SPANNING-TREE is the following decision problem:

   Given an undirected graph G and an integer k, does G have a spanning tree with k or fewer leaves?

   Is this problem in **P** or is it **NP**-Complete? Justify your answer.

   (Hint: Is the problem more closely related to MST, the decision version of the minimum spanning tree problem, or to HAMILTONIAN-CYCLE (i.e., does the graph have a simple cycle through all of the vertices? You can assume we know HAMILTONIAN-CYCLE is **NP**-Complete.)

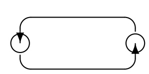
6. Consider an undirected graph G with source vertices $s_1, s_2, \ldots s_k$ and sink vertices $t_1, t_2, \ldots, t_k$. The NETWORK-ROUTING decision problem asks whether there are k node disjoint paths where the ith path goes from s_i to t_i. Show that this problem is **NP**-Complete.

Hints: Reduce form 3-SAT. For a 3-SAT formula with q clauses and n variables, use $k = q + n$ sources and sinks. Introduce one source/sink pair (s_x, t_x) for each variable x, and one source/sink pair (s_c, t_c) for each clause c. Now, for each 3-SAT clause, introduce (at most) 6 new intermediate vertices, one for each literal occurring in that clause and one for the negation fo that literal. Now finish this off by noticing that if the path from s_c to t_c goes through a vertex representing a variable x, then no other path can go through that vertex. What other vertex would you like that path to go through instead?

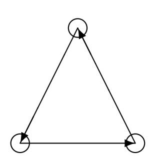
7. (EXTRA CREDIT – please really try this!) Consider a directed graph $G = (V, E)$. A *kernel* for graph G is a subset K of the vertices such that:

   - Every vertex in V is either in the kernel K or has an incoming edge from some vertex in the kernel. (Formally, for all $v \in V, v \in K$ or there exists $u \in K$ such that $(u, v) \in E$.)
   - There do not exist two vertices in the kernel K with a directed edge between them.

   Observe that the graph below has a kernel, namely either of the two vertices:

   On the other hand, the graph below has no kernel.

   Use these two little graphs to show that deciding whether a graph has a kernel is **NP**-Complete.