1. (C-12.8) Let $S$ be a set of $n$ intervals of the form $[a,b]$ where $a < b$. Design an efficient data structure that can answer, in no more than $O(\log^2 n + k)$ time, queries of the form contains$(x)$, which asks for a list of all intervals in $S$ which contain $x$. $k$ is the number of such intervals. What is the space usage of your data structure? [Notice that other than finding a place to begin searching, you are not allowed to paw through lots of intervals that aren't ones you select. If this wasn't a concern, lots of things would work.]

2. Jennie and Ben have found a treasure chest. They have a bag of valuables and want to divide it up. For each of the following scenarios, either give a polynomial-time algorithm to divide up the money, or prove that the problem is NP-complete. The input in each case is a list of $n$ items in the bag, along with their value. **If the problem is polynomial, you demonstrate that by showing me a polynomial time algorithm to solve it. If the problem is NP-complete, you demonstrate that fact with a mapping from a known NP-complete problem.**
   a. There are $n$ coins, but only 2 different denominations: Gobbles and Hortz. Gobbles are worth $x$ dollars, and Hortz are worth $y$ dollars. They wish to divide the coins so the money is divided exactly evenly.
   b. There are $n$ coins with $m$ different denominations, but each denomination is a nonnegative integer power of 2, i.e., the possible denominations are 1 dollar, 2 dollars, 4 dollars, etc. They wish to divide the money exactly evenly.
   c. There are $n$ checks, which are, amazingly enough, all made out to "Cash." They wish to divide the checks (not just cash them all and divide the money) so that they each get the exact same amount of money.
   d. There are $n$ checks as in part c, but this time Alice and Bob are willing to accept a split in which the difference is no greater than 100 dollars.

3. CNF is a difficult problem. Is disjunctive normal form also difficult? DNF is the “or” of a bunch of “anded” terms. Show that the problem of determining the satisfiability of boolean formulas in disjunctive normal form is polynomial time solvable or show it is NP-complete. **Hint: Don't go on instinct here. Really figure it out.**

4. 2CNF-SAT is the set of satisfiable boolean formulas in CNF with exactly two literals per clause. For example, $(a+b)(c+d)(\neg b+c)$. Show the 2CNF-SAT $\in$ P. (Remember you show membership in P by demonstrating an algorithm.) Make your algorithm as efficient as possible.

5. (C-13.6) Suppose a friend has given you a magic computer $C$ such that when given any Boolean formula $B$ in CNF it will tell you in one step if $B$ is satisfiable or not. Show how to use $C$ to construct an actual assignment of satisfying boolean values to the variables in the formula. How many calls to $C$ do you need to make?

6. (C-13/8) Define an Independent-Set as the problem that takes a graph $G$ and an integer $k$ and asks if $G$ contains an independent set of vertices of size $k$. That is, $G$ contains a set $I$ of vertices of size $k$ such that for any $v$ and $w$ in $I$, there is no edge $(v,w)$ in $G$. Show that the Independent-Set problem is NP complete. (Be sure to show both that it is in NP and that it is difficult.)

7. (C-13.11) Show that the Hamiltonian-Cycle problem on directed graphs is NP-complete. **Hint: Notice this is the directed version you are being asked to solve. This is actually pretty easy. You are allowed to map from any of the other NP-complete problems (page 604)**