Aim to do at least five of the eight problems. Be sure to pick one with some dynamic programming component. When describing a DP solution, describe carefully the substructure and a recurrence for it before giving the algorithm. Also describe the time and space complexity of your algorithm.

1. Suppose we have a collection of \( n \) semi-encrypted files, where semi-encrypted means that we can decode each one, but it is pretty slow to do so. However, we do have the capability to quickly determine if any two files were created by the same person - we call this procedure \( \text{eq}(f_1, f_2) \). Give an algorithm to determine whether at least \( n/2 \) files were created by the same person, using at most \( O(n \lg n) \) calls to procedure \( \text{eq} \).

2. Show how to find the smallest vertex cover of a tree \( T \) in linear time.

3. You are driving to a conference a very long ways away. Your grant allows you to stay in hotels at distances \( a_1 < a_2 < \cdots < a_n \) from your starting point, and the final hotel is where the conference is being held. Ideally, you want to drive 500 miles a day - if you drive \( x \) miles in a day, your penalty for that day is \( (500 - x)^2 \). Design an algorithm to decide at which hotels to stop in such a way as to minimize the total penalty.

4. exercise 21-3, p 521

5. Consider the Exact Sum (ES) problem: given integers \( s_1, s_2, \ldots, s_n \) and an integer \( K \), does a subset of \( s_1, s_2, \ldots, s_n \) add up to \( K \)?

   (a) Show that ES is NP-complete.

   (b) Using dynamic programming, give an \( O(nK) \) algorithm to find such a subset, if it exists.

   (c) Modify the problem to allow any \( s_i \) to be used multiple times - this is the Exact Sum with Repetition (ESR) problem. Equivalently, we want to know whether we can make change for \( K \) cents using coins denominated \( s_1, s_2, \ldots, s_n \). Give a dynamic programming solution for ESR.

   (d) Is ESR NP-complete?

6. In class we outlined but did not complete a proof that \( 2\text{SAT} \in P \). Complete the proof by carefully describing an algorithm for \( 2\text{SAT} \).

7. Show that \( P^{NP} \cap coNP = NP \cap coNP \).

8. Given a set \( A = \{a_1, a_2, \ldots, a_n\} \) and a collection \( B_1, B_2, \ldots, B_m \) of subsets of \( A \) (that is, each \( B_i \subseteq A \)), a hitting set \( H \subseteq A \) has the property that it contains at least one element from every \( B_i \). In other words, \( \forall i, B_i \cap H \neq \emptyset \). The Hitting Set (HS) problem is to determine if there is a hitting set of size at most \( k \), for a given integer \( k \). Show that HS is NP-complete.
Notes:

- (Q1) Divide and conquer!
- (Q2&3) Dynamic programming.
- (Q5) I’m sure ES has another name, but this way you can’t look it up. Part (c) is basically exercise 16-1(d), p 402. I don’t know the answer to part (d), so your guess is acceptable.
- (Q8) Garey & Johnson suggest using the vertex cover problem.