Problem 1: Suppose you are trying to navigate a set of roads to get from point A to point B. Assume that the roads are on a grid. All roads are one-way roads, and point either straight north or straight east. Your job is to get from the southwest corner to the northeast corner of the grid. At each intersection, there are traffic lights, each of which has some delay associated with it. Let $d_{i,j}$ represent the delay caused by the intersection at position (i, j) of the grid. At each intersection, you can turn in the direction of the intersecting road, or continue on the current road. Design a dynamic programming algorithm to determine the path of roads you should take to get from the southwest corner to the northeast corner of the grid with the smallest total delay. Assume you don't need to deal with delays at your starting or ending point.

Hint: Define $T[i, j]$ to be the minimum total delay required to get to the (i, j) intersection from the starting position. What subproblems do you need in order to calculate this value?

Problem 2: A palindrome is a sequence of characters that is the same if it is reversed. For example, ‘anna’ is a palindrome. Suppose that you are given a sequence of characters, and wish to determine whether it is made of one of more palindromes placed next to each other. For example, ‘dadseesanna’ is made of the palindromes ‘dad’, ‘sees’, and ‘anna’, placed next to each other. Design a dynamic programming algorithm to solve this problem. Hint: It may be useful to write a helper function to determine whether a particular string is a palindrome.

Problem 3: Suppose that you are trying to find a max flow in a directed graph. However, instead of having capacities on the *edges*, you have capacities on the *nodes*. (Can you think of a real-life application for this kind of problem?) Show how to find a max flow in such a case. Hint: like we did in class, think about how to modify the graph, rather than the algorithm.

Problem 4: There are n animals who are sick and need to see a doctor. These animals are of different species: there are some horses, some dogs, some cats, some goats, etc. Each animal a_i must be seen for a total of h_i hours: for example, one animal might require surgery that takes 10 hours, while another animal just has a cold and only needs to be seen for 1 hour. It is ok to split this time between as many doctors as you need: for instance, in the 10 hour surgery, one doctor can do 1 hour of the surgery, another could do 7, and a third could do 2.

There are m doctors, and each doctor is able to treat certain species of animals, but not necessarily all (for example, some doctors might be able to treat dogs and cats, but not horses, goats, or elephants). Additionally, each doctor D_j has a cap C_j on the total number of hours that she is able to work. Assume all h_i and C_j values are positive integers. Design an algorithm to determine how to assign doctors to animals so that each animal is seen for the required number of hours.

Problem 5: We are organizing a computer science conference, and scientists from around the world have submitted a total of N papers. M people have volunteered to act as reviewers. Each paper needs to be reviewed by 3 different people. Some of the people who submitted papers are also reviewers, and we need to make sure that no person is assigned to review their own papers. Each reviewer R_i is willing to review some maximum number m_i of papers. Design an algorithm to determine how to assign papers to reviewers.

Extra Credit: You have a set of N boxes, where each box B_i has m_i pieces of fruit. Each box contains a specific type of fruit- e.g., bananas, apples, mangos- and no two boxes have the same type of fruit. Design a dynamic programming algorithm to determine how many ways are there to select M fruits (e.g., if you are trying to select 5 fruits, an option would be 2 bananas and 3 apples, or 4 mangos and 1 banana). You can select at most m_i pieces of fruit from box B_i.