CSCI 270: Homework 6

1. From the lecture, you know how to use dynamic programming to solve the 0-1 knapsack problem where each item is unique and only one of each kind is available. Now let us consider the knapsack problem where you have infinitely many items of each kind. Namely, there are n different types of items. All the items of the same type i have equal size wi and value vi. You are offered with infinitely many items of each type. Design a dynamic programming algorithm to compute the optimal value you can get from a knapsack with capacity W.

    a. Define (in plain English) subproblems to be solved. (4 pts)
    b. Write a recurrence relation for the subproblems (6 pts)
    c. Make sure you specify
        a. base cases and their values (2 pts)
        b. where the final answer can be found (1 pt)
    d. What is the complexity of your solution? (2 pts)

2. Solve Kleinberg and Tardos, Chapter 6, Exercise 12.

    a. Define (in plain English) subproblems to be solved. (4 pts)
    b. Write a recurrence relation for the subproblems (6 pts)
    c. Make sure you specify
        i. base cases and their values (2 pts)
        ii. where the final answer can be found (1 pt)
    d. What is the complexity of your solution? (2 pts)

3. Given n balloons, indexed from 0 to $n - 1$. Each balloon is painted with a number on it represented by array nums. You are asked to burst all the balloons. If the you burst balloon i you will get $nums[left] * nums[i] * nums[right]$ coins. Here $left$ and $right$ are adjacent indices of i. After bursting the balloon, the $left$ and $right$ then becomes adjacent. You may assume $nums[-1] = nums[n] = 1$ and they are not real therefore you can not burst them. Design a dynamic programming algorithm to find the maximum coins you can collect by bursting the balloons wisely. Analyze the running time of your algorithm.

    Here is an example. If you have the nums arrays equals $[3, 1, 5, 8]$. The optimal solution would be 167, where you burst balloons in the order of 1, 5, 3 and 8. The left balloons after each step is:

    $$[3, 1, 5, 8] \rightarrow [3, 5, 8] \rightarrow [3, 8] \rightarrow [8] \rightarrow []$$

And the coins you get equals:

$$(3 * 1 * 5) + (3 * 5 * 8) + (1 * 3 * 8) + (1 * 8 * 1) = 167$$

a. Define (in plain English) subproblems to be solved. (4 pts)
b. Write a recurrence relation for the subproblems (6 pts)
c. Make sure you specify
    i. base cases and their values (2 pts)
    ii. where the final answer can be found (1 pt)
d. What is the complexity of your solution? (2 pts)

4. Suppose you have a rod of length N, and you want to cut up the rod and sell the pieces in a way that maximizes the total amount of money you get. A piece of length i is worth p_i dollars. Devise a Dynamic Programming algorithm to determine the maximum amount of money you can get by cutting the rod strategically and selling the cut pieces.

    a. Define (in plain English) subproblems to be solved. (4 pts)
    b. Write a recurrence relation for the subproblems (6 pts)
    c. Using the recurrence formula in part b, write pseudocode to solve the problem. (5 pts)
    d. Make sure you specify
        i. base cases and their values (2 pts)
        ii. where the final answer can be found (1 pt)
    e. What is the complexity of your solution? (2 pts)

5. Solve Kleinberg and Tardos, Chapter 6, Exercise 10.
    a. part(a) of the question (4pts)
    b. part(b) of the question (answer according to the format below)
        i. Define (in plain English) subproblems to be solved. (4 pts)
        ii. Write a recurrence relation for the subproblems (6 pts)
        iii. Using the recurrence formula in part ii, write pseudocode to solve the problem. (5 pts)
        iv. Make sure you specify
            i. base cases and their values (2 pts)
            ii. where the final answer can be found (1 pt)
        v. What is the complexity of your solution? (2 pts)