# CSCI 570 - Fall 2022 - HW 6

Due Oct. 12th

**Note**: This homework assignment covers dynamic programming. It is recommended that you read chapter 6.1 to 6.4 from Kleinberg and Tardos.

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 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 $w_i$ and value $v_i$. 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 $W$.

2. Given a non-empty string $s$ and a dictionary containing a list of unique words, design a dynamic programming algorithm to determine if $s$ can be segmented into a space-separated sequence of one or more dictionary words. If $s$ = *"algorithmdesign"* and your dictionary contains *"algorithm"* and *"design"*. Your algorithm should answer Yes as $s$ can be segmented as *"algorithmdesign"*.

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

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 pi 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.

5. Solve Kleinberg and Tardos, Chapter 6, Exercise 6.
6. Solve Kleinberg and Tardos, Chapter 6, Exercise 10.