



Dynamic Programming Assignment

1. On a positive integer, you can perform any one of the following 3 steps. 1.) Subtract 1 from it. ($n = n - 1$), 2.) If its divisible by 2, divide by 2. (if $n \% 2 == 0$, then $n = n / 2$), 3.) If its divisible by 3, divide by 3. (if $n \% 3 == 0$, then $n = n / 3$). Now the question is, given a positive integer n , find the minimum number of steps that takes n to 1 eg:
 - a. For $n = 1$, output: 0
 - b. For $n = 4$, output: 2 ($4 / 2 = 2 / 2 = 1$)
 - c. For $n = 7$, output: 3 ($7 - 1 = 6 / 3 = 2 / 2 = 1$)
2. Given a string find the longest substring which is a palindrome.
3. Coin Change: Given a list of N coins, their values (V_1, V_2, \dots, V_N), and the total sum S . Find the minimum number of coins the sum of which is S (we can use as many coins of one type as we want), or report that it's not possible to select coins in such a way that they sum up to S .
4. Given weights and values of n items, put these items in a knapsack of capacity W to get the maximum total value in the knapsack. In other words, given two integer arrays $val[0..n-1]$ and $wt[0..n-1]$ which represent values and weights associated with n items respectively. Also given an integer W which represents knapsack capacity, find out the maximum value subset of $val[]$ such that sum of the weights of this subset is smaller than or equal to W . You cannot break an item, either pick the complete item, or don't pick it (0-1 property).