|  |  |
| --- | --- |
| **WEEK-14** | **DYNAMIC PROGRAMMING** |
| **1.** For a given string **S,** write a program based on dynamic programming for finding the length of the longest substring of S which contains exactly K distinct vowels. Analyze the complexity of the proposed algorithm.  **Sample Input & Output**  Input: s = “artyebui”, k = 2  Output: 6  Explanation: Longest substring with only 2 vowel is “rtyebu”  **2.** Given weights and values of n items, put these items in a knapsack of capacity M, to get the maximum total value in the knapsack. Note that- there are infinite instances of each item available. So, any item can be selected any number of times. Analyze the complexity of the proposed algorithm.  **Sample Input and Output:**  Number of items, n=3  Weight of items, w[]=4 8 2  Value of items, v[]=7 8 3  Capacity of knapsack, M=11  Maximum attainable value of items=17  (by collecting first item twice and third item once)  **3.** A contiguous subsequence of a list S is a subsequence made up of consecutive elements of S. For instance, if S is 5; 15; -30; 10; -5; 40; 10; then 15; -30; 10 is a contiguous subsequence but 5; 15; 40 is not.  Given a list of numbers, write a linear-time algorithm for the finding the contiguous subsequence of maximum sum.  **Sample Input & Output**  Input: 5; 15; -30; 10; -5; 40; 10;  Output: 10;-5; 40; 10, with a sum of 55.  **4.** Given a set of integers, Write an efficient program to find out whether or not this set can be divided into two subsets such that the sum of elements in each set is equal.  **5.** There are two strings ***src***and ***dest.*** Write a program to convert ***src***to ***dest***by applying minimum edits operation on the string *src*. The edit operations are as following:  a) Insert a character  b) Delete a character  c) Replace a character  **Sample Input & Output** | |