Practice Problems in Data Structures and Algorithms

NO NEED TO SUBMIT

1. Design an algorithm to find three elements within an array which sum to a specified value. You may assume the elements in the array to be both distinct and non-distinct.
2. You are given an array of integers (both positive and negative). Find the continuous sequence with the largest sum. Return the sum
   EXAMPLE
   Input: {2, -8, 3, -2, 4, -10}
   Output: 5 (i.e., {3, -2, 4})
3. Write an algorithm which computes the number of trailing zeros in n factorial
4. You are given two sorted arrays, A and B, and A has a large enough buffer at the end to hold B. Write a method to merge B into A in sorted order.
5. Given a sorted array of n integers that has been rotated an unknown number of times, give an O(log n) algorithm that finds an element in the array. You may assume that the array was originally sorted in increasing order.
   EXAMPLE:
   Input: find 5 in array (15 16 19 20 25 1 3 4 5 7 10 14)
   Output: 8 (the index of 5 in the array)
6. Given a sorted array of strings which is interspersed with empty strings, write a method to find the location of a given string.
   Example: find “ball” in [“at”, “”, “”, “”, “ball”, “”, “”, “car”, “”, “”, “dad”, “”, “”] will return 4
   Example: find “ballcar” in [“at”, “”, “”, “”, “”, “ball”, “car”, “”, “”, “dad”, “”, “”] will return -1
7. Given a matrix in which each row and each column is sorted, write a method to find an element in it.
8. Write a method to generate the nth Fibonacci number.
9. Write a method that returns all subsets of a set.
10. Write a method to compute all permutations of a string.
11. Write an algorithm to find the ‘next’ node (i.e., in-order successor) of a given node in a binary search tree where each node has a link to its parent.
12. Implement a function to check if a tree is balanced. For the purposes of this question, a balanced tree is defined to be a tree such that no two leaf nodes differ in distance from the root by more than one.
13. Design an algorithm to find the lowest common ancestor of two nodes in a binary tree. Avoid storing additional nodes in a data structure. NOTE: This is not necessarily a binary search tree.
14. You are given a binary tree in which each node contains a value. Design an algorithm to print all paths which sum up to that value. Note that it can be any path in the tree - it does not have to start at the root.
15. Write a method to remove duplicates from an unsorted linked list.
16. Implement an algorithm to determine if a string has all unique characters. What if you cannot use additional data structures?
17. Design an algorithm to remove the duplicate characters in a string without using an additional buffer.