Advanced Algorithmic Problem Solving (R1UC601B)

PRACTICE QUESTIONS FOR MTE

1. What is meant by time complexity and space complexity? Explain in detail.
2. What are asymptotic notations? Define Theta, Omega, big O, small omega, and small o.
3. Explain the meaning of O(2^n), O(n^2), O(nlgn), O(lg n). Give one example of each.
4. Explain sliding window protocol.
5. Explain Naive String-Matching algorithm. Discuss its time and space complexity.

6 Explain Rabin Karp String-Matching algorithm. Discuss its time and space complexity.
7 Explain Knuth Morris and Pratt String-Matching algorithm. Discuss its time and space complexity.
8 What is a sliding window? Where this technique is used to solve the programming problems.
9 What are bit manipulation operators? Explain and, or, not, ex-or bit-wise operators.
10 Why are the benefits for linked list over arrays.
11 Implement singly linked list.
11 Implement stack with singly linked list.
12 Implement queue with singly linked list.
12 Implement doubly linked list.
13 Implement circular linked list.
14 Implement circular queue with linked list.
15 What is recursion? What is tail recursion?
16 What is the tower of Hanoi problem? Write a program to implement the Tower of Hanoi problem. Find the time and space complexity of the program.
17 What is backtracking in algorithms? What kind of problems are solved with this technique?
18 Implement N-Queens problem. Find the time and space complexity.
19 What is subset sum problem? Write a recursive function to solve the subset sum problem?
20 Implement a function that uses the sliding window technique to find the maximum sum of any contiguous subarray of size K.
21 Write a recursive function to generate all possible subsets of a given set.
22 Write a program to find the first occurrence of repeating character in a given string.
23 Write a program to print all the LEADERS in the array. An element is a leader if it is greater than all the elements to its right side. And the rightmost element is always a leader.
24 Write a program to find the majority element in the array. A majority element in an array A[] of size n is an element that appears more than n/2 times.
25 Given an integer k and a queue of integers, write a program to reverse the order of the first k elements of the queue, leaving the other elements in the same relative order.
26 Write a program to implement a stack using queues.
27 Write a program to implement queue using stacks.
28 Given a string S of lowercase alphabets, write a program to check if string is

isogram or not. An Isogram is a string in which no letter occurs more than once.

29 Given a sorted array, arr[] consisting of N integers, write a program to find the frequencies of each array element.
30 Write a program to delete middle element from stack.
31 Write a program to remove consecutive duplicates from string.
33 Write a program to display next greater element of all element given in array.
34 Write a program to evaluate a postfix expression.
35 Write a program to get MIN at pop from stack.
36 Write a program to swap k^{th} node from ends in given single linked list.
37 Write a program to detect loop in linked list
38 Write a program to find Intersection point in Y shaped Linked list.
39 Write a program to merge two sorted linked list.
40 Write a program to find max and second max of array.
41 Write a program to find Smallest Positive missing number. You are given an array arr[] of N integers. The task is to find the smallest positive number missing from the array. Positive number starts from 1.
42 Given a non-negative integer N. The task is to check if N is a power of 2. More formally, check if N can be expressed as 2x for some integer x. Return true if N is power of 2 else return false.
43 Write a program to Count Total Digits in a Number using recursion. You are given a number n. You need to find the count of digits in n.
44 Check whether K-th bit is set or not. Given a number **N** and a bit number **K**, check if **Kth** index bit of **N** is set or not. A bit is called set if it is 1. Position of set bit '1' should be indexed starting with 0 from LSB side in binary representation of the number. Index is starting from 0. You just need to return **true** or **false**.
45 Write a program to print 1 To N without loop.