IMPORTANT QUESTIONS AC2

1. Explain Insertion at the random position and count number of nodes in singly linked list with pseudo codes?
2. Advantages of linked list over array? Write a function to implement insertion and deletion at the end DLL?
3. Explain Queue, its applications and implement all functionalities of Queue using linked list?
4. Define Queue and implement circular queue using array (enqueue and dequeue)?
5. Write a code implement the stack using queue?
6. Explain in detail about quick sort technique and analyse the time and space complexity of quick sort?
7. Illustrate the merge sort with example and write the code snippet?
8. Explain about Binary search algorithm and discus it's time and space complexities?
9. what is collision in hashing and explain different collision resolution techniques with examples?
10. Explain binary tree, Binary search tree and different types of binary tree?
11. Implement tree Data structure and pre-order, In-order, post order traversal of binary tree using recursion?
12. Illustrate different ways to represent the graph?
13. Compute BFS, DFS for the given Graph?
14. Construct the pre order tree for the given post order and in order traversal(take your example)
15. Explain about the job scheduling problem?
16. Explain about the Dijkstra's Algorithm?
17. Explain how a Fractional knapsack algorithm works with suitable example and pseudo code?
18. Explain KMP algorithm with pseudo code and suitable examples?
19. Explain about the Rabin-Karp algorithm?
20. What is Dynamic Programming? when to use Dynamic Programming? What are the approaches in Dynamic Programming?

Problems to Practice:

1. The stock span problem is a financial problem where we have a series of **n** daily price quotes for a stock and **write a program** to **calculate** the span of stocks price for all **n** days. The span S_i of the stock's price on a given day **i** is defined as the maximum number of consecutive days just before the given day, for which the price of the stock on the given day is less than or equal to its price on the current day.
   Example:
   Input:
   N = 7, price[] = [100 80 60 70 60 75 85]
   Output:
   1 1 1 2 1 4 6
2. Write a program to reverse the given linked list without extra space?
3. Given an array **height** representing the heights of buildings. You have to count the buildings that will see the sunrise (Assume the sun rises on the side of the array starting point). *Note:* The height of the building should be strictly greater than the height of the buildings left in order to see the sun.
   Example:
   Input: height = [7, 4, 8, 2, 9]

Output: 3

Explanation: As 7 is the first element, it can see the sunrise. 4 can't see the sunrise as 7 is hiding it. 8 can see. 2 can't see the sunrise. 9 also can see the sunrise.

4. You are given an array of characters letters that is sorted in **non-decreasing order**, and a character target. There are **at least two different** characters in letters.

   print the smallest character in letters that is lexicographically greater than target. If such character does not exist, print the first character in letters.

   Example :

   Input: letters = ["c","f","j"], target = "a"

   Output: "c"

   Explanation: The smallest character that is lexicographically greater than 'a' in letters is 'c'.

5. Given a non-negative integer x, print the square root of x rounded down to the nearest integer. The resultant integer should be **non-negative** as well.

   You **must not use** any built-in exponent function or operator.

   For example, do not use pow(x, 0.5) in c++.

   Example:

   Input: x = 8

   Output: 2

   Explanation: The square root of 8 is 2.82842..., and since we round it down to the nearest integer, 2 is returned.

6. You are given an integer **mountain** array arr of length n where the values increase to a **peak element** and then decrease. Write a program to print the index of the peak element.

   Your task is to solve it in O(log(n)) time complexity.

   Example 1:

   Input: arr = [0,2,5,8,12,7,6,4] n=8

   Output: 4

7. You are given an array of characters letters that is sorted in **non-decreasing order**, and a character target. There are **at least two different** characters in letters.

   print the smallest character in letters that is lexicographically greater than target. If such a character does not exist, print the first character in letters.

   Example :

   Input: letters = ["c","f","j"], target = "a"

   Output: "c"

   Explanation: The smallest character that is lexicographically greater than 'a' in letters is 'c'.

8. Write a code to find the count of the leaf nodes in given binary tree

   In given input **N** represents Null.

   Example 1:

   Input: 1 2 3

```
  1
 / \
2   3
```

   1. **Output**: 2

   Example 2:

   Input: 2 N 1 3 N

```
2
 \
  1
 /
3
```

   Output:1

9. Given a sorted array of distinct integers and a target value, the index if the target is found. If not, return the index where it would be if it were inserted in order.

   You must implement it with O(log n) runtime complexity.

Example 1:
Input: nums = [1,3,5,6], target = 5
Output: 2
Example 2:
Input: nums = [1,3,5,6], target = 2
Output: 1

10. At a lemonade stand, each lemonade costs $5. Customers are standing in a queue to buy from you and order one at a time (in the order specified by bills). Each customer will only buy one lemonade and pay with either a $5, $10, or $20 bill. You must provide the correct change to each customer so that the net transaction is that the customer pays $5.

Note that you do not have any change in hand at first.

Given an integer array bills where bills[i] is the bill the ith customer pays, print true ***if you can provide every customer with the correct change, or*** **false** ***otherwise.***

Example :
Input: bills = [5,5,5,10,20]
Output: true
Explanation:
From the first 3 customers, we collect three $5 bills in order.
From the fourth customer, we collect a $10 bill and give back a $5.
From the fifth customer, we give a $10 bill and a $5 bill.
Since all customers got correct change, we output true.

11. Given a binary array nums, write a program to print *the maximum number of consecutive* 1*'s in the array.*

Example:
Input: nums = [1,1,0,1,1,1]
Output: 3
Explanation: The first two digits or the last three digits are consecutive 1s. The maximum number of consecutive 1s is 3.

12. There are **n** gas stations along a circular route, where the amount of gas at the ith station is gas[i].You have a car with an unlimited gas tank and it costs cost[i] of gas to travel from the ith station to its next (i + 1)th station. You begin the journey with an empty tank at one of the gas stations.Given two integer arrays **gas** and **cost**, print the starting gas station's index if you can travel around the circuit once in the clockwise direction, otherwise print -1. If there exists a solution, it is guaranteed to be unique.

Input: gas = [1,2,3,4,5], cost = [3,4,5,1,2]
Output: 3
Explanation:
Start at station 3 (index 3) and fill up with 4 unit of gas.
Your tank = 0 + 4 = 4
Travel to station 4. Your tank = 4 - 1 + 5 = 8
Travel to station 0. Your tank = 8 - 2 + 1 = 7
Travel to station 1. Your tank = 7 - 3 + 2 = 6
Travel to station 2. Your tank = 6 - 4 + 3 = 5
Travel to station 3. The cost is 5. Your gas is just enough to travel back to station 3. Therefore, return 3 as the starting index

13. Regional's CS Department is writing a spell-checker system, and you have been tasked with writing a function to determine how closely two words resemble each other. The algorithm you are to use, albeit not a very good one, is to compare the two words character by character, and count how many times the characters in a given position are the same.

You are given Strings **A** and **B** and you have to print an integer **K** indicating the score (as defined above) of how closely the two match.

Examples:

Input: A=" TICK" B=" TOCK " **Output**: 3

Input: A=" DOG" B=" GOD" **Output**: 1

14. Given a text **txt** and a pattern **pat**, write a function that finds all occurrences of **p** and its permutations (or anagrams) in **t**. You may assume n > m where n is length of text and m is length of pattern.

Examples:

***Input**: t = "BACDGABCDA", p = "ABCD"*

***Output**: [0, 5, 6]*

Explanation *: "BACD" is at 0, "ABCD" at 5 and "BCDA" at 6*

15. Given two strings **s** and **goal**, print true *if and only if* **s** *can become* **goal** *after some number of* ***shifts*** *on* **s**.A **shift** on **s** consists of moving the leftmost character of **s** to the rightmost position.

Example:

Input: s = "abcde", goal = "cdeab"

Output: true

16. Given a list of 24-hour clock time points in **"HH:MM"** format, print the minimum **minutes** difference between any two time-points in the list.

Example 1:

Input: timePoints = ["23:59","00:00"]

Output: 1

Example 2:

Input: timePoints = ["00:00","23:59","00:00"]

Output: 0

17. You are given an array prices where prices[i] is the price of a given stock on the i^{th} day.

You want to maximize your profit by choosing a **single day** to buy one stock and choosing a **different day in the future** to sell that stock.

Compute *the maximum profit you can achieve from this transaction*. If you cannot achieve any profit, print 0.

Example 1:

Input: prices = [7,1,5,3,6,4]

Output: 5

Explanation: Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = 6-1 = 5.

Note that buying on day 2 and selling on day 1 is not allowed because you must buy before you sell.

18. There are *N* stones, numbered 1,2,…,N. For each *i* (1≤i≤N), the height of Stone *i* is h_i.

There is a frog who is initially on Stone 1. He will repeat the following action some number of times to reach Stone N: If the frog is currently on Stone i, jump to Stone i+1 or Stone i+2 Here, a cost of $|h_i−h_j|$ is incurred, where j is the stone to land on.

Write a program to find the minimum possible total cost incurred before the frog reaches Stone N.
Example:
Input:
4
10 30 40 20
Output:
30

19. Given a fence with **n** posts and **k** colors, Write a program to find out the number of ways of painting the fence so that **not more than two** consecutive posts have the same colors.
Example:
Input:
n = 3
k = 2
Output: 6
Explanation:
Let the 2 colours be 'R' and 'B'.
We have following possible combinations:
1. RRB
2. RBR
3. RBB
4. BRR
5. BRB
6. BBR

20. You are given an integer array cost where cost[i] is the cost of i^{th} step on a staircase. Once you pay the cost, you can either climb one or two steps.
You can either start from the step with index 0, or the step with index 1.
Print the minimum cost to reach the top of the floor.
Example:
Input: cost = [1,100,1,1,1,100,1,1,100,1]
Output: 6
Explanation: You will start at index 0.
- Pay 1 and climb two steps to reach index 2.
- Pay 1 and climb two steps to reach index 4.
- Pay 1 and climb two steps to reach index 6.
- Pay 1 and climb one step to reach index 7.
- Pay 1 and climb two steps to reach index 9.
- Pay 1 and climb one step to reach the top.
The total cost is 6.