**Q1** Suppose we apply Huffman coding to an alphabet of size 4, and the resulting tree is a perfectly balanced binary tree (one root with two children, each of which has two children of its own). Find the maximum frequency of any letter.

**Q2** Describe the backtracking approach to algorithm design. Give an example of a problem that can be solved using this technique.

**Q16** There is a bi-directional graph with n vertices, where each vertex is labeled from 0 to n - 1 (inclusive). The edges in the graph are represented as a 2D integer array edges, where each edges[i] = [ui, vi] denotes a bi-directional edge between vertex ui and vertex vi. Every vertex pair is connected by at most one edge, and no vertex has an edge to itself.

You want to determine if there is a valid path that exists from vertex source to vertex destination.

Given edges and the integers n, source, and destination, return true if there is a valid path from source to destination, or false otherwise.

Example 1:

Input: n = 3, edges = [[0,1],[1,2],[2,0]], source = 0, destination = 2

Output: true

Explanation: There are two paths from vertex 0 to vertex 2:

- 0 → 1 → 2

- 0 → 2

**Q3** Explain in brief about the Backtracking approach for algorithm design. How it differs with recursion?

**Q17** Explain the N-Queen Problem and its algorithm using backtracking and analyze its time complexity

**Q18** What is dynamic programming? Find the longest common subsequence between “XYYXXY” and “XXYXXYY”.

**Q4** Explain the Union and Find operation in algorithm design. Write a program to detect a cycle in directed graph

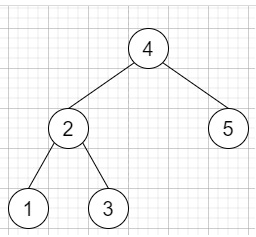
**Q5** Implement topological sorting

**Q6** Define AVL tree. Construct AVL tree from given data set: 4, 6, 12, 9, 5, 2, 13, 8, 3, 7, 11.

Q7 You are provided with balanced binary tree with the target value k. return x number of values that are closest to the given target k. provide solution in O(n)

Note: You have only one set of unique values x in binary search tree that are closest to the target.

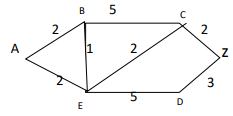
Input:



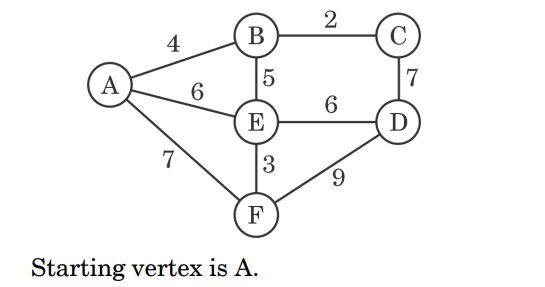
K=3.8, x=2

Output: 3,4

**Q8** Trace and implement Kruskal’s Algorithm to find minimum spanning tree for the following graph.



**Q9** What is hashing? Explain concept of hash table and hash function with example and Implement Linear probing collision resolution function

**Q10** Write Kruskal algorithm to implement minimum spanning tree. Trace below graph to find minimum spanning tree using Kruskal algorithm.

**[ 5 Marks]**

**Q11** What is a binary search tree? what is the maximum number of nodes possible in a

binary tree of depth d? make a BST for the following sequence of numbers and traverse

tree in all types of traversals.

45,32,90,21,78,65,87,132,90,96,41,74,92

**Q12** Given the root of a binary tree with unique values and the values of two different nodes of the tree x and y, return true if the nodes corresponding to the values x and y in the tree are brothers, or false otherwise.

Two nodes of a binary tree are brothers if they have the same depth with different parents.

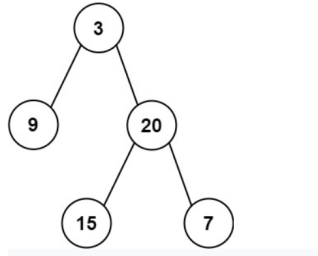
Note that in a binary tree, the root node is at the depth 0, and children of each depth k node are at the depth k + 1.

Input: root = [1,2,3,4], x = 4, y = 3

Output: false

**Q13** Is hashing better than binary search algorithm? Give reasons. Define any two collision resolution techniques.

**Q14** Given the root of a binary tree, write an algorithm to return *the sum of all left leaves.* A **leaf** is a node with no children. A **left leaf** is a leaf that is the left child of another node.



**Output: 24 {sum of left node: 9+15=24}**

**Q15**

You are provided list of single bitcoin price on each day, return maximum amount one could have lost by buying or selling bitcoin each day.

input: {200,300,100,700,300,300,100}

output:600

buying on day 1(price 300) selling on day 2(price 100)

buying on day 3(price 700) selling on last day(price 100)