
Question 1 : How to reverse a String in java? Can you write a program without using any java inbuilt methods?

Solution: There are many ways to do it, some of them are:

- Using for loop
- Using recursion
- Using StringBuffer

Question 2 : Write a java program to check if two Strings are anagram in java?

Solution: Two string are anagrams if they have same characters but in different order. For example: Angel and Angle are anagrams
There are few ways to check if Strings are anagrams. Some of them are:

. 1) Using String methods
. 2) Using array.sort

Question 3 : Write a program to check if String has all unique characters in java?

Solution: Here are some ways to check if String contains all unique characters

- By using HashSet
- Using indexOf and lastIndexOf methods of String
- By Using ascii value of characters.

Question 4 : How to check if one String is rotation of another String in java?

Solution: Let's say you want to check whether str1 and str2 is rotation of one another or not.

-> Create a new String with str3= str1 + str1

-> Check if str3 contains str2 or not.

-> if str3 contains str2 then str2 is rotation of str1 else it is not

Question 5 : How to find duplicate characters in String

in java?

Solution: Here is a solution to find duplicate

characters in String.

. Create a HashMap and character of String will be inserted as key and its count as value.

. If Hashamap already contains char,increase its count by 1, else put char in HashMap.

. If value of Char is more than 1, that means it is duplicate character in that String.

Question 6 : Find first non repeated character in String in java

Solution: There are may ways to

find it. Some of them are:

- Using LinkedHashMap
- Using indexOf and lastIndexOf methods.

Question 7 : Find all substrings of String in java?

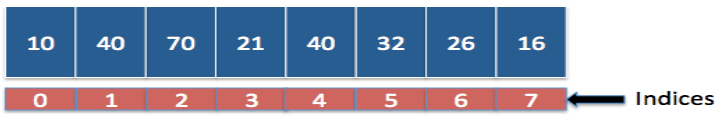
Solution: Java program to find all substrings of a String.
For example: If input is “abb” then output should be “a”, “b”,”b”, “ab”, “bb”, “abb”

We will use String class’s subString method to find all subString.

Question 8 : Find length of String without using any inbuilt method in java?

Solution: You can use try catch block for catching StringIndexOutOfBoundException and when this exception aries, you can simply return i(Index at which you will get the exception)

Question 9 : Write a program to print all permutations of String in java?



Array of length 8

Solution: Take out first character of String and insert into different places of permutations of remaining String recursively.

Question 10 : Write java Program to Find Smallest and Largest Element in an Array.

You are given an integer array containing n element. You need to provide an optimum solution to find the smallest and largest number. Number can not be repeated.

Question 11 : Find missing number in the array.

You are given an integer array containing 1 to n but one of the number from 1 to n in the array is missing. You need to provide optimum solution to find the missing number. Number cannot be repeated in the array.

Question 12 : Search an element in rotated and sorted array.

You are given an sorted and rotated array as below:

```
int arr[]={16,19,21,25,3,5,8,10};
```

If you note that array is sorted and rotated. You need to search an element in above array in o(log n) time complexity.

Question 13 : Find minimum element in a sorted and rotated array.

You are given an sorted and rotated array as below:

```
int arr[]={16,19,21,25,3,5,8,10};
Minimum element in the
array : 3
```

If you note that array is sorted and rotated. You need to i an element in above array in o(log n) time complexity.

Question 14: Find second largest number in an array

You are given an sorted and rotated array as below:

For example:

```
int[] arr1={7,5,6,1,4,2};
Second largest element in  the
array : 6
```

Question 15 : Find the number occurring odd number of times in an array

You are given a array of integer. All numbers occur even number of times except one. You need to find the number which occurs odd number of time. You need to solve it with o(n) time complexity and o(1) space complexity.
For example:

```
int array[] = new int[]{20, 40, 50, 40, 50, 20, 30, 30, 50, 20, 40, 40, 20};
Number which occurs odd  number of times
is : 50
```

Question 16 : Find minimum number of platforms required for railway station

You are given arrival and departure time of trains reaching to a particular station. You need to find minimum number of platforms required to accommodate the trains at any point of time.

For example:

```
 arrival[] = {1:00, 1:40, 1:50, 2:00, 2:15, 4:00}
departure[] = {1:10, 3:00, 2:20, 2:30, 3:15, 6:00}
No. of platforms required in above
scenario = 4
```

Please note that arrival time is in chronological order.

Question 17 : Find a Pair Whose Sum is Closest to zero in Array

Given array of +ve and -ve integers ,we need to find a pair whose sum is closed to Zero in Array.

For example:

```
array[]={1,3,-5,7,8,20,-40,6};
The pair whose sum is closest to zero :  -
5 and 6
```

Question 18 : Given a sorted array and a number x, find the pair in array whose sum is closest to x

---

Given a sorted array, we need to find a pair whose sum is closed to number X in Array.

For example:

```
array[]={-40,-5,1,3,6,7,8,20};
The pair whose sum is closest to 5 :
1and 3
```

Question 19 : Find all pairs of elements from an array whose sum is equal to given Number

Given a array,we need to find all pairs whose sum is equal to number X.

For example:

```
array[]={ -40, -5, 1, 3, 6, 7, 8, 20 };
Pair of elements whose sum is equal to 15 : 7, 8 and -5, 20
```

Question 20: Given an array of 0's and 1's in random order, you need to separate 0's and 1's in an array.

For example:

```
arr[] = {0,1,0,0,1,1,1,0,1}
Array after separating 0 and 1 numbers :
{0,0,0,0,1,1,1,1,1}
```

Question 21 : Separate odd and even numbers in an array

Given an array of integers, you need to segregate odd and even numbers in an array. Please note: Order of elements can be changed.

For example:

```
arr[] = {12, 17, 70, 15, 22, 65, 21, 90}
Array after separating odd and even
numbers : {12, 90, 70, 22, 15, 65, 21, 17}
```

Question 22 : Given an array containing zeroes, ones and twos only. Write a function to sort the given array in O(n) time complexity.

For example:

Input :
[1, 2, 2, 0, 0, 1, 2, 2, 1]

Output :
[0, 0, 1, 1, 1, 2, 2, 2, 2]

Question 23 : Find local minima in array

A local minima is less than its neighbours

For example:

Input :

```
int [] arr = {10, 5, 3, 6, 13, 16, 7};
```

Output: 2

```
int []arr = {11,12,13,14};
```
Output: 11

```
int []arr = {10};
```
Output: 10

```
int []arr = {8,6};
```
Output: 6

Question 24 : Sliding window maximum in java

Given an Array of integers and an Integer k, Find the maximum element of from all the contiguous subarrays of size K.

For example:

Input :
Input : int[] arr = {2,6,-1,2,4,1,-6,5}
int k = 3
output : 6,6,4,4,4,5

Question 25 : Count number of occurrences (or frequency) of each element in a sorted array

Given a Sorted Array of integers containing duplicates. Find the frequency of every unique element present in the array.
Frequency is defined as the number of occurrence of any element in the array.

For example :

```
Input:
int[] arr = {1, 1, 1, 3, 3, 4, 5, 5, 6, 6};
Output:
Frequency  of 1 is :  3
Frequency  of 3 is :  2
Frequency  of 4 is :  1
Frequency  of 5 is :  2
Frequency  of 6 is :  2
```

Question 26 : Find subarrays with given sum in an array.

Given an Array of non negative Integers and a number. You need to print all the starting and ending indices of Subarrays having their sum equal to the given integer.

For example :

```
Input :
Input-int[] arr = {2, 3, 6, 4, 9,
0,11};

int num = 9
Output-

starting index : 1,
Ending index : 2
starting index : 5,
Ending index : 5
starting index : 5,
Ending  index  : 6
```

Question 27 : Find peak element in the array.

Peak Element is the element of the array which is GREATER THAN / EQUAL

TO its neighbours, that is, for an element at i th index, the neighbour elements at index i-1 & i+1 must be greater than equal to element at i th position.

Question 28 : Find leaders in an array.

We need to print all the leaders present in the array. Element is the leader if it is greater than right side of elements.

arr[]={14, 12, 70, 15, 99, 65, 21, 90}
Here 99 **and** 90 are leader
elements

Question 29 : Count 1's in sorted Binary Array.

Print number of 1's in a given sorted Binary Array.

For example :

Input :
int[] arr = {0,0,0,1,1,1,1};
output : 4
int[] arr = {0,0,1};
output : 1

Question 30 : Find first repeating element in an array of integers.

Find the first repeating element in array of integers.

For example :

Input :
Input: **array**[] = {10, 7, 8, 1, 8, 7, 6}
Output: 7 [7 **is** the first element actually
repeats]

Question 31 : Check if Array Elements are Consecutive.

Given an array, we need to check if array contains consecutive elements.

For example :

---

input: **array**[] = {5, 3, 4, 1, 2}
Output: **true**
As array contains consecutive elements from 1 **to** 5
Input: **array**[] = {47, 43, 45, 44, 46}
Output: **true**
As array contains consecutive elements from 43 **to** 47
Input: **array**[] = {6, 7, 5, 6}
Output: **false**
As array does **not** contain consecutive
elements.

Question 32 : Permutations of array in java.

Given array of distinct integers, print all permutations of the array.

For example :

array : [10, 20, 30]
permuations are :

[10, 20, 30]
[10, 30, 20]
[20, 10, 30]
[20, 30, 10]
[30, 10, 20]
[30, 20, 10]

Question 33 : Rotate an array by K positions.

For example :

N=6 **and** k=2
If Arr[] = {1, 2, 3, 4, 5, 6} **and** k=2
then rotated **array** will be {5, 6, 1, 2, 3, 4}

Question 34 : Stock Buy Sell to Maximize Profit.

Given an array of integers representing stock price on single day, find max profit that can be earned by 1 transaction.

So you need to find pair (buyDay,sellDay) where buyDay < = sellDay and it should maximise the profit.

For example :

```
int arr[]={14, 12, 70, 15, 99, 65, 21, 90};
Max profit can be gain by buying at 1th day(0 based indexing) and sell
At 4th day. 4Max profit = 99-12 =87
```

Question 35 : Find maximum difference between two elements such that larger element appears after the smaller number.

Given array of integers, find Maximum difference between two elements such that larger element appears after the smaller number

For example :

```
int arr[]={14, 12, 70, 15, 95, 65, 22, 30};
Max Difference =95-12 = 83
```

Question 36 : Search in a row wise and column wise sorted matrix.

Given row wise and column wise sorted matrix ,we need to search element with minimum time complexity.

Question 37 : Largest sum contiguous subarray.

Largest sum contiguous subarray is the task of finding the contiguous subarray within a one-dimensional array of numbers which has the largest sum.

For example :

```
for the sequence of values −2, 1, −3, 4, −1, 2, 1, −5, 4; the contiguous subarray with t
the largest sum is 4, −1, 2, 1, with sum 6
```

Question 38 : Find the Contiguous Subarray with Sum to a Given Value in an array.

Given an array of positive integer and given value X, find Contiguous sub array whose sum is equal to X.

For example :

```
arr[]={14, 12, 70, 15, 99, 65, 21, 90};
X =97.
Sum found between index 1 to 3
Elements are 12, 17 and 15
```

Question 39 : Longest Common Prefix in an array of Strings in java.

Given an array of positive integer and given value X, find Contiguous sub array whose sum is equal to X.

For example :

```
String[] strArr={"java2blog","javaworld","javabean","javatemp"};
So Longest common prefix in above String array will be “java”  as all above  string  star  4ts with “java”.
```

Question 40 : Find all subsets of set (power set) in java.

Given a set of distinct integers, arr, return all possible subsets (the power set).

For example :

```
Input: nums = [1,2,3]
Output:
[
  [3],
  [1],
  [2],
  [1,2,3],
  [1,3],
  [2,3],
  [1,2],
  []
]
```

Question 41: Implement a stack using array.

You need to implement Stack using array. You need to write push and pop methods to demonstrate Stack behavior(Last In First Out).

Question 42: Implement a stack using Linked List.

You need to implement Stack using Linked List. You need to write push and pop methods to demonstrate Stack behavior(Last In First Out).

Question 43: Implement a stack using two queues.

You need to use two queues to implement stack behavior.You need to write push and pop methods to demonstrate Stack behavior(Last In First Out).

Question 44 : Sort an stack using another stack

You need to sort an stack using another stack. You can use push and pop operation of stack to do so,

Question 45: Implement Queue using Array in java.

You need to use array to implement queue.

Question 46: Implement a stack using two queues .

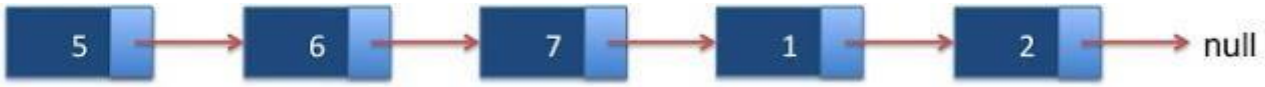
You need to use Linked list to implement queue.

Question 47 : Implement singly linked list in java.

You need to implement singly linked list data structures.You need to write

---

simple program to demonstrate insert , delete operations.

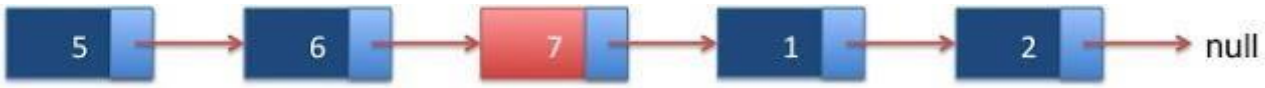


Question 48: How to reverse linked list in java.

You need to write iterative and recursive solution to reverse linked list.

Question 49: How to find middle element of linked list.

You need to write java program to find middle element of linked list in most optimize way.



Question 50 : How to find nth element from end of linked list .

You need to write java program to find nth element of linked list in most optimize way. In question 6, Node 7 is 3rd from last of linked list.

Question 51 : How to detect a loop in linked list. If linked list has loop, find the start node for the loop.

You need to write a java program to detect whether any loop exists in linked list and if loop exists , you need to find start node for the linked list.

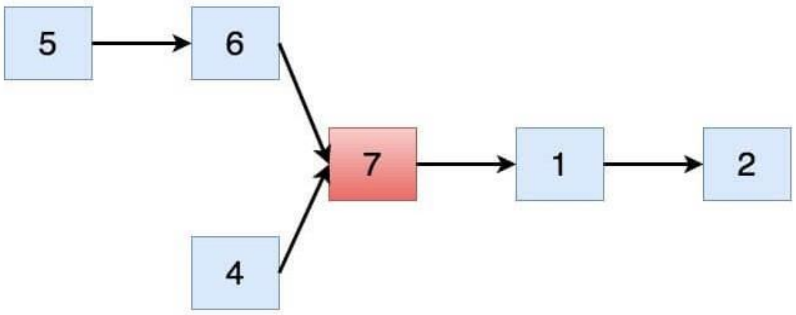
Question 52: How to check if linked list is palindrome or not?

A palindrome is a word, phrase, number, or other sequence of symbols or elements that reads the same forward or reversed. For example: 12121 is palindrome as it reads same forward or

reversed. madam is also a palindrome . So we need write java programs to check if linked list is palindrome or not.

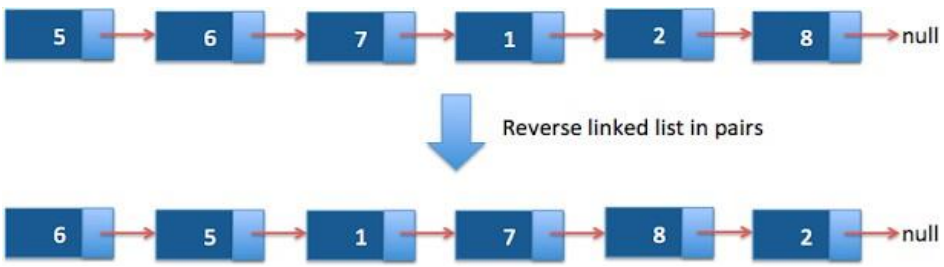
Question 53 : Find intersection of two linked lists?

Given two singly linked lists, find if two linked lists intersect. If they intersect, find intersection point.



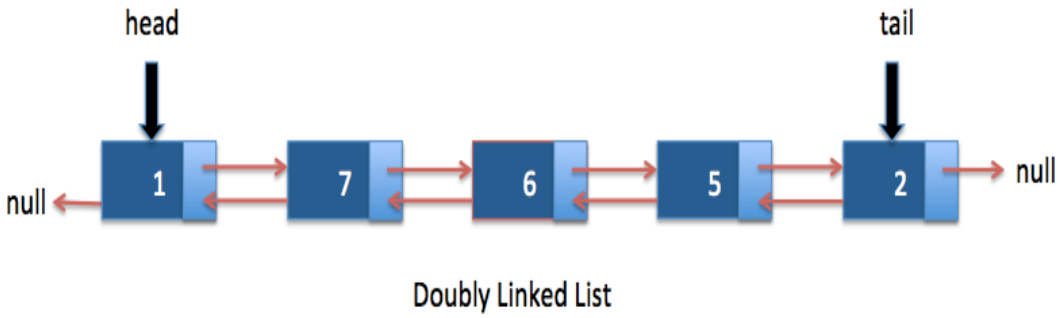
Question 54 : How to reverse a linked list in pairs?

You need to write a java program to reverse linked list in pairs.



Question 55 : Implement Doubly linked list in java?

You need to write a java program to implement doubly linked list in java.



Doubly Linked List

Question 56 : How can you traverse binary tree?

There are three ways to traverse binary tree.

- PreOrder
- InOrder
- PostOrder.

Question 57 : Write an algorithm to do level order traversal of binary tree?

You need to write java program to do level order traversal of binary tree. You can use queue data structure to do level order traversal.

Question 58 : Write an algorithm to do spiral order traversal of binary tree?

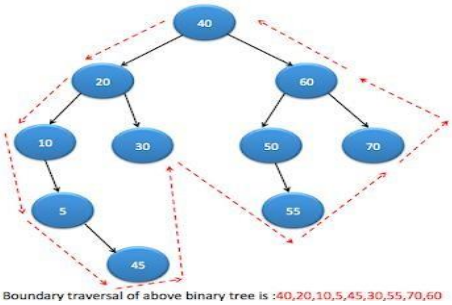
You need to write java program to do spiral level order traversal of binary tree

Question 59 : How can you print leaf nodes of binary tree?

You need to write java program to print all leaf nodes of binary tree.

Question 60 : How to count leaf nodes of binary tree.

You need to write java program to count leaf nodes of binary tree.



Question 61 : How to print all paths from root to leaf in binary tree.

You need to write a program to print all paths from root to leaf.

Question 62 : How to find level of node in binary tree

Given a node, you need to find level of a node. For example : Level of node will 3 for node 70 used in Question 14.

Question 63 : How to find maximum element in binary tree.

You need to write a java program to find maximum element in binary tree.

Question 64 : How to find lowest common ancestor(LCA) in binary tree.

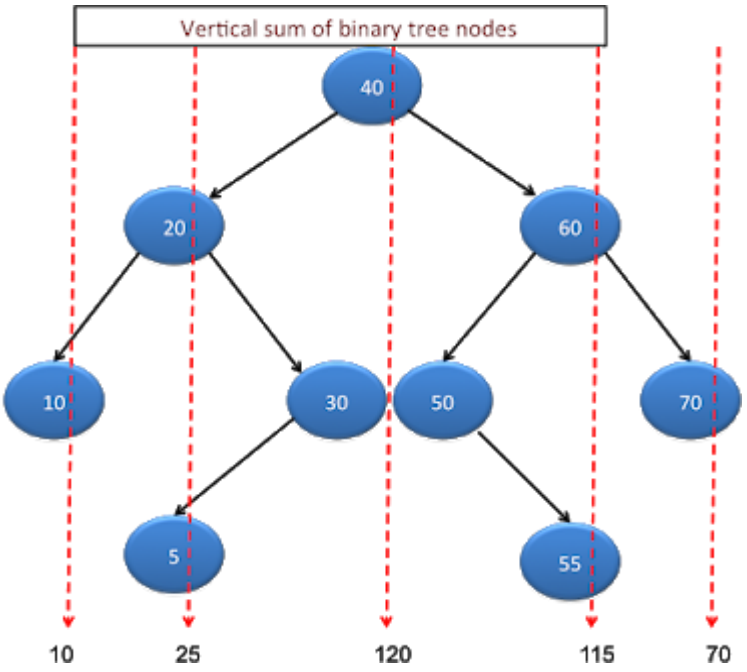
You need to write a program to find LCA in binary tree.

Question 65 : How to do boundary traversal of binary tree.

Write a java program to do boundary traversal of binary tree.

Question 66 : How to print vertical sum of binary tree?

You need to find sum of nodes which lies in same column.



Question 67 : Count subtrees with Sum equal to target in binary tree?

Given a Binary tree and an integer. You need to find the number of subtrees having the sum of all of its nodes equal to given Integer, that is, Target sum.

Question 68 : What is binary search tree?

Binary search tree is a special type of binary tree which have following properties.

- Nodes which are smaller than root will be in left subtree.
- Nodes which are greater than root will be right subtree.
- It should not have duplicate nodes
- Both left and right subtree also should be binary search tree.

Question 69 : Can you write algorithm to insert a node in binary search tree.

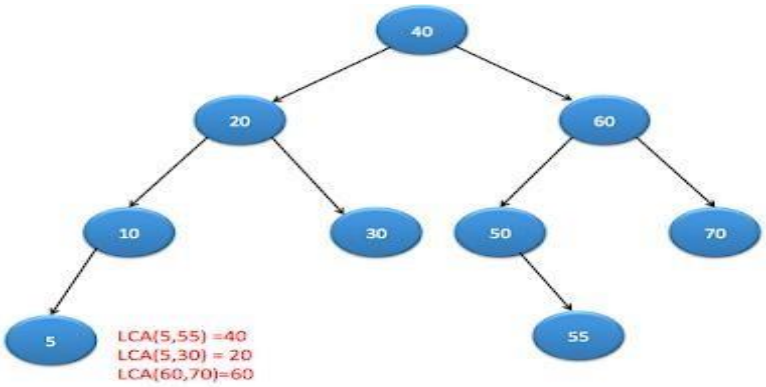
Question 70 : Can you write algorithm to delete a node in binary search tree.

Question 71 : How can you find minimum and maximum elements in binary search tree?

Solution : Leftmost and rightmost nodes of binary search tree are minimum and maximum nodes respectively

Question 72 : How to find lowest common ancestor(LCA) in binary search tree.

You need to write a program to find LCA in binary search tree.



Question 73 : Find inorder successor in a Binary search Tree

You need to write a program to find inorder successor in a Binary search tree.

Question 74 : Convert sorted array to balanced BST

Question 75 : Convert sorted Linked List to balanced BST

Question 76 : Check if a binary tree is binary search tree or not in java

Question 77 : Write an algorithm to implement bubble sort?

Question 78 : Write an algorithm to implement insertion sort sort?

Question 79 : Write an algorithm to implement selection sort sort?

Question 80 : Can you write algorithm for merge sort and also do you know complexity of merge sort?

Question 81 : Do you know how to implement Heap sort?

Question 82 : Implement quick sort in java?

Question 83 : Implement shell sort in java?

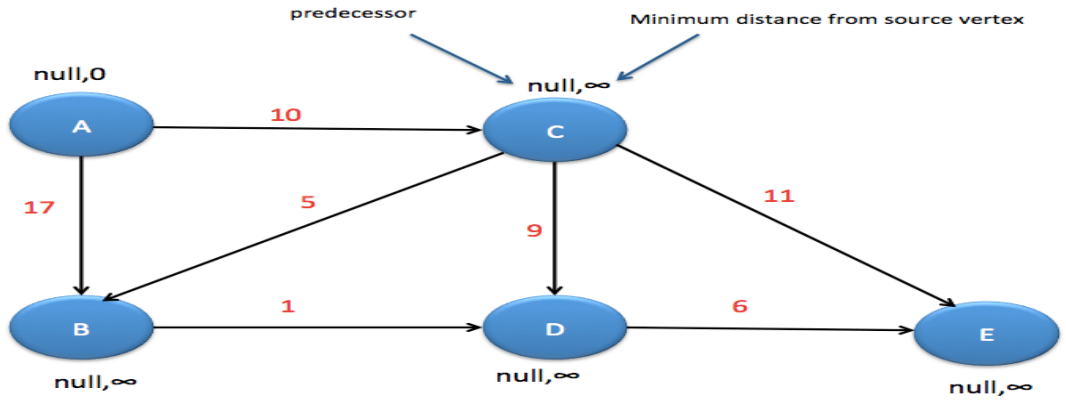
Question 84 : Implement Counting sort in java?

Question 85 : What is binary search? Can you write an algorithm to find an element in sorted array using binary search?

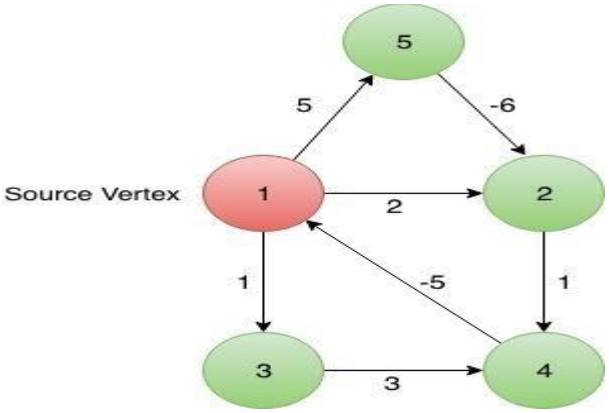
Question 86 : Write algorithm to do depth first search in a graph.

Question 87 : Write algorithm to do breadth first search in a graph.

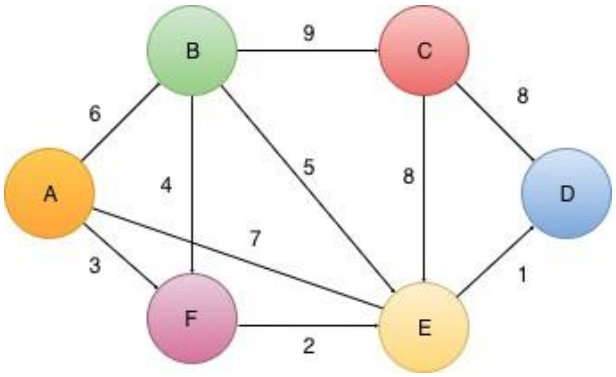
Question 88 : Explain Dijkstra algorithm from source to all other vertices.



Question 89 : Explain Bellman Ford algorithm to find shortest distance



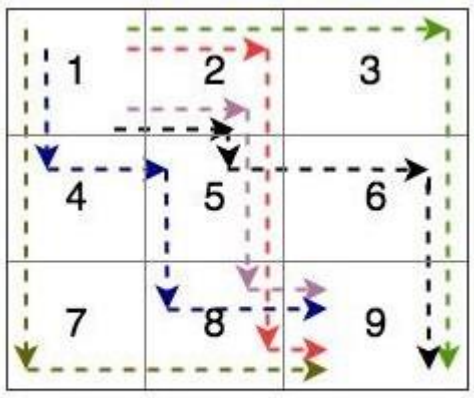
Question 90 : Explain Kruskal's algorithm for finding minimum spanning tree



Question 91 : Given two String, find longest common substring.

Question 92 : Given two Strings A and B. Find the length of the Longest Common Subsequence (LCS) of the given Strings.

Question 93 : Given a matrix, we need to count all paths from top left to bottom right of MxN matrix. You can either move down or right.



No. of paths from top left to bottom right : 6

Given two strings string1 and string2, String1 is to be converted into String2 with the given operations available in the minimum number of steps. Using any one of the given operations contributes to the increment of steps by one.

Allowed Operations are :

(i) **Remove** : This operation allows the Removal any one character from String.

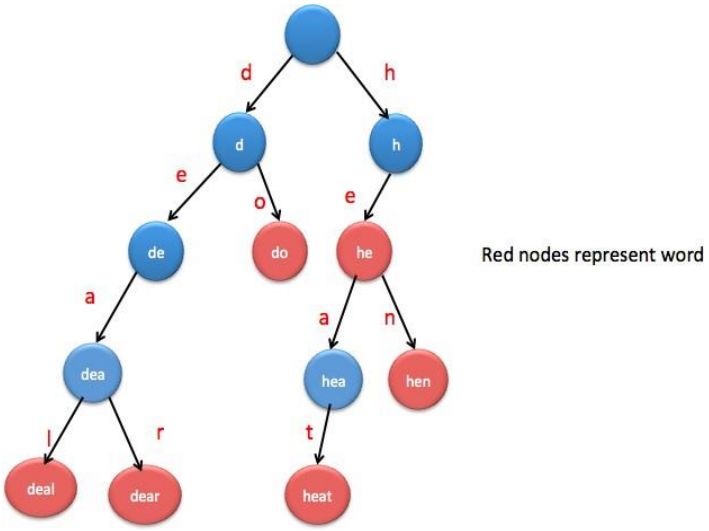
(ii) **Insert** : This operation allows the Insertion of one character at any spot in the String.

(iii) **Replace** : This operation allows the replacement of any one character in the string with any other character.

Given an Amount to be paid and the currencies to pay with. There is infinite supply of every currency using combination of which, the given amount is to be paid. Print the number of ways by which the amount can be paid.

Question 97 : What is an algorithm and how to calculate complexity of algorithms.

Question 98 : Implement trie data structure in java.



Trie Data structure

Question 99 : Count Factorial Trailing Zeroes in java.

Question 100 : Largest Rectangular Area in a Histogram.

Question 101 : Check for balanced parentheses in an expression in java.
