

Java Coding Interview

Table of Contents [[hide](#)]

- [String](#)

- [Question 1 : How to reverse a String in java? Can you write a program without using any java inbuilt methods?](#)
- [Question 2 : Write a java program to check if two Strings are anagram in java?](#)
- [Question 3 : Write a program to check if String has all unique characters in java?](#)
- [Question 4 : How to check if one String is rotation of another String in java?](#)
- [Question 5 : How to find duplicate characters in String in java?](#)
- [Question 6 : Find first non repeated character in String in java?](#)
- [Question 7 : Find all substrings of String in java?](#)
- [Question 8 : Find length of String without using any inbuilt method in java?](#)
- [Question 9 : Write a program to print all permutations of String in java?](#)

- [Array](#)

- [Question 10 : Write java Program to Find Smallest and Largest Element in an Array.](#)
- [Question 11 : Find missing number in the array.](#)
- [Question 12 : Search an element in rotated and sorted array.](#)
- [Question 13 : Find minimum element in a sorted and rotated array.](#)
- [Question 14: Find second largest number in an array](#)

- [Question 15 : Find the number occurring odd number of times in an array](#)
- [Question 16 : Find minimum number of platforms required for railway station](#)
- [Question 17 : Find a Pair Whose Sum is Closest to zero in Array](#)
- [Question 18 : Given a sorted array and a number x, find the pair in array whose sum is closest to x](#)
- [Question 19 : Find all pairs of elements from an array whose sum is equal to given number](#)
- [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.](#)
- [Question 21 : Separate odd and even numbers in an array](#)
- [Question 22 : Given an array containing zeroes, ones and twos only. Write a function to sort the given array in O\(n\) time complexity.](#)
- [Question 23 : Find local minima in array](#)
- [Question 24 : Sliding window maximum in java](#)
- [Question 25 : Count number of occurrences \(or frequency\) of each element in a sorted array](#)
- [Question 26 : Find subarrays with given sum in an array.](#)
- [Question 27 : Find peak element in the array.](#)
- [Question 28 : Find leaders in an array.](#)
- [Question 29 : Count 1's in sorted Binary Array.](#)
- [Question 30 : Find first repeating element in an array of integers.](#)
- [Question 31 : Check if Array Elements are Consecutive.](#)
- [Question 32 : Permutations of array in java.](#)
- [Question 33 : Rotate an array by K positions.](#)
- [Question 34 : Stock Buy Sell to Maximize Profit.](#)

- [Question 35 : Find maximum difference between two elements such that larger element appears after the smaller number.](#)
- [Question 36 : Search in a row wise and column wise sorted matrix.](#)
- [Question 37 : Largest sum contiguous subarray.](#)
- [Question 38 : Find the Contiguous Subarray with Sum to a Given Value in an array.](#)
- [Question 39 : Longest Common Prefix in an array of Strings in java.](#)
- [Question 40 : Find all subsets of set \(power set\) in java.](#)

- [Stack](#)

- [Question 41: Implement a stack using array.](#)
- [Question 42: Implement a stack using Linked List.](#)
- [Question 43: Implement a stack using two queues.](#)
- [Question 44 : Sort an stack using another stack](#)

- [Queue](#)

- [Question 45: Implement Queue using Array in java.](#)
- [Question 46: Implement a stack using two queues .](#)

- [Linked List](#)

- [Question 47 : Implement singly linked list in java.](#)
- [Question 48: How to reverse linked list in java.](#)
- [Question 49: How to find middle element of linked list.](#)
- [Question 50 : How to find nth element from end 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.](#)
- [Question 52: How to check if linked list is palindrome or not?](#)
- [Question 53 : Find intersection of two linked lists?](#)
- [Question 54 : How to reverse a linked list in pairs?](#)
- [Question 55 : Implement Doubly linked list in java?](#)

- Binary Tree

- Question 56 : How can you traverse binary tree?
- Question 57 : Write an algorithm to do level order traversal of binary tree?
- Question 58 : Write an algorithm to do spiral order traversal of binary tree?
- Question 59 : How can you print leaf nodes of binary tree?
- Question 60 : How to count leaf nodes of binary tree.
- Question 61 : How to print all paths from root to leaf in binary tree.
- Question 62 : How to find level of node in binary tree
- Question 63 : How to find maximum element in binary tree.
- Question 64 : How to find lowest common ancestor(LCA) in binary tree.
- Question 65 : How to do boundary traversal of binary tree.
- Question 66 : How to print vertical sum of binary tree?
- Question 67 : Count subtrees with Sum equal to target in binary tree?

- Binary Search tree

- Question 68 : What is 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?
- Question 72 : How to find lowest common ancestor(LCA) in binary search tree.
- Question 73 : 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](#)

- [Sorting](#)

- [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?](#)

- [Graph](#)

- [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](#)

- [Dynamic Programming](#)

- [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.](#)
- [Question 94 : Edit Distance Problem in java](#)
- [Question 95: Coin change problem in java](#)
- [Question 96 : Minimum number of jumps to reach last index](#)
- [Miscellaneous](#)
 - [Question 97 : What is an algorithm and how to calculate complexity of algorithms.](#)
 - [Question 98 : Implement trie data structure in java.](#)
 - [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.](#)
 - [Question 102 : What is Memoization.](#)

If you want to practice and improve [data structure](#) and algorithm programs, this post will be very helpful to you. I will recommend you to try it yourself first and then check the solution.

String

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

Please refer to the solution at [reverse a String in java](#)

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

Check solution at [check if two Strings are anagram in java.](#)

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.

Please refer to complete solution at [check if String has all unique 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.

1. Create a new String with `str3= str1 + str1`
2. Check if str3 [contains](#) str2 or not.
3. if str3 [contains](#) str2 then str2 is rotation of str1 else it is not

You can find complete solution at [check if one String is rotation of another in java](#).

Question 5 : How to find duplicate characters in String in java?

Solution: Here is a solution to find duplicate characters in String.

1. Create a [HashMap](#) and character of String will be inserted as key and its count as value.
2. If [Hashamap](#) already contains char, increase its count by 1, else put char in HashMap.
3. If value of Char is more than 1, that means it is duplicate character in that String.

Please refer to solution at [program to find duplicate characters in a String](#).

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

Solution: There are many ways to find it.

Some of them are:

- Using [LinkedHashMap](#)
- Using indexOf and lastIndexOf methods.

Please find complete solution at [find first non repeated character in a String](#).

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.

Please refer to complete solution at [find all subStrings of String](#).

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

Solution: You can use try catch block for catching

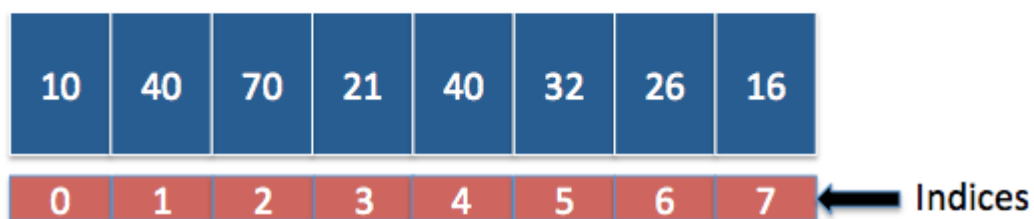
StringIndexOutOfBoundsException and when this exception arises, you can simply return i (Index at which you will get the exception)

Please refer to complete solution at [find length of String without inbuilt methods](#).

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

Solution: Take out first character of String and insert into different places of permutations of remaining String recursively. Please find complete solution at [how to find all permutations of String in java](#).

Array



Array of length 8

You may be asked lot of java coding interview questions on Array. You can practice following coding questions on Array to ace coding interview.

Question 10 : Write java Program to Find Smallest and Largest Element in an 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 an optimum solution to find the missing number. Number can not be repeated in the array.

For example:

```
1
2 int[] arr1={7,5,6,1,4,2};
3 Missing number : 3
4 int[] arr2={5,3,1,2};
5
```

6Missing numner : 4

Solution : [Java Program to Find Smallest and Largest Element in an Array](#)

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.

For example:

```
1
2int[] arr1={7,5,6,1,4,2};
3Missing numner : 3
4int[] arr2={5,3,1,2};
5Missing numner : 4
6
```

Solution : [Find missing number in the array.](#)

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

You are given an sorted and rotated array as below:

```
1
2int arr[]={16,19,21,25,3,5,8,10};
3
```

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

Solution : [Search element in rotated and sorted array](#)

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

You are given an sorted and rotated array as below:

```
1
2: int arr[]={16,19,21,25,3,5,8,10};
3: Minimum element in the array : 3
4
```

If you note that array is sorted and rotated. You need to find an element in above array in $O(\log n)$ time complexity.

Solution : [Find minimum element in a sorted and rotated array](#)

Question 14: Find second largest number in an array

You are given an sorted and rotated array as below:

For example:

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

Solution : [java program to find second largest number in an array.](#)

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:

```
1
```

```
2 int array[] = new int[]{20, 40, 50, 40, 50, 20, 30, 30, 50, 20, 40, 40, 20};
```

```
3 Number which occurs odd number of times is : 50
```

```
4
```

Solution : [java program to find number occurring odd number of times in an array.](#)

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:

```
1
```

```
2 arrival[] = {1:00, 1:40, 1:50, 2:00, 2:15, 4:00}
```

```
3 departure[] = {1:10, 3:00, 2:20, 2:30, 3:15, 6:00}
```

```
4
```

```
5 No. of platforms required in above scenario = 4
```

Please note that arrival time is in chronological order.

Solution : [Find minimum number of platforms required for railway station.](#)

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:

```
1
```

2 `array[]={1,3,-5,7,8,20,-40,6};`

3 The pair whose sum is closest to zero : -5 and 6

4

Solution : [Find a Pair Whose Sum is Closest to zero in Array in java.](#)

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:

1

2 `array[]={-40,-5,1,3,6,7,8,20};`

3 The pair whose sum is closest to 5 : 1 and 3

4

Solution : [Find a Pair Whose Sum is Closest to X in Array in java.](#)

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:

1

2 `array[]={ -40, -5, 1, 3, 6, 7, 8, 20 };`

3 Pair of elements whose sum is equal to 15 : 7, 8 and -5, 20

4

Solution : [Find all pairs of elements from an array whose sum is equal to given number .](#)

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:

```
1
2 arr[] = {0,1,0,0,1,1,0,1}
3 Array after separating 0 and 1 numbers :
4 {0,0,0,0,1,1,1,1}
5
```

Solution : [Separate 0s and 1s in array.](#)

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:

```
1
2 arr[] = {12, 17, 70, 15, 22, 65, 21, 90}
3 Array after separating odd and even numbers :
4 {12, 90, 70, 22, 15, 65, 21, 17}
5
```

Solution : [Separate 0s and 1s in array.](#)

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:

```
1
```

```
2 Input :
3 [1, 2, 2, 0, 0, 1, 2, 2, 1]
```

```
5 Output :
6 [0, 0, 1, 1, 1, 2, 2, 2, 2]
```

Solution : [Sort an array of 0s, 1s and 2s.](#)

Question 23 : Find local minima in array

A local minima is less than its neighbours

For example:

```
1
2 Input :
3
4 int [] arr = {10, 5, 3, 6, 13, 16, 7};
```

```
5 Output: 2
```

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

```
9 Output: 11
```

```
11 int []arr = {10};
```

```
12 Output: 10
```

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

```
15 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:

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

Solution : [Find the local minima in array.](#)

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 :

```
1
2 Input :
3 Input:
4 int[] arr = {1, 1, 1, 3, 3, 4, 5, 5, 6, 6};
5 Output:
6 Frequency of 1 is : 3
7 Frequency of 3 is : 2
8
```

9 Frequency of 4 is : 1

10 Frequency of 5 is : 2

11 Frequency of 6 is : 2

Solution : [Count number of occurrences \(or frequency\) of each element in a sorted array.](#)

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 :

```
1
2 Input :
3 Input-int[] arr = {2, 3, 6, 4, 9, 0, 11};
4 int num = 9
5 Output-
6 starting index : 1, Ending index : 2
7 starting index : 5, Ending index : 5
8 starting index : 5, Ending index : 6
9
```

Solution : [Find subarrays with given sum in an array.](#)

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.

Solution : [Find peak element in the array.](#)

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.

```
1
2 arr[]={14, 12, 70, 15, 99, 65, 21, 90}
3 Here 99 and 90 are leader elements
4
```

For example:

Solution : [Find leaders in an array.](#)

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

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

For example :

```
1
2 Input :
3 int[] arr = {0,0,0,1,1,1,1};
4 output : 4
5 int[] arr = {0,0,1};
6 output : 1
7
```

Solution : [Count 1's in sorted Binary Array.](#)

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

Find the first repeating element in array of integers.

For example :

```
1
2 Input :
3 Input: array[] = {10, 7, 8, 1, 8, 7, 6}
4 Output: 7 [7 is the first element actually repeats]
5
```

Solution : [Find first repeating element in an array of integers.](#)

Question 31 : Check if Array Elements are Consecutive.

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

For example :

```
1
2 Input: array[] = {5, 3, 4, 1, 2}
3 Output: true
4 As array contains consecutive elements from 1 to 5
5 Input: array[] = {47, 43, 45, 44, 46}
6 Output: true
7 As array contains consecutive elements from 43 to 47
8 Input: array[] = {6, 7, 5, 6}
9 Output: false
10 As array does not contain consecutive elements.
11
```

Solution : [Check if Array Elements are Consecutive.](#)

Question 32 : Permutations of array in java.

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

For example :

```
1
2 array : [10, 20, 30]
3
4 Permuations are :
5
6 [10, 20, 30]
7
8 [10, 30, 20]
9
10 [20, 10, 30]
11
12 [20, 30, 10]
13
14 [30, 10, 20]
15
16 [30, 20, 10]
```

Solution : [Permutations of array in java.](#)

Question 33 : Rotate an array by K positions.

For example :

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

Solution : [Rotate an array by K positions.](#)

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 :

```
1
2 int arr[]={14, 12, 70, 15, 99, 65, 21, 90};
3 Max profit can be gain by buying at 1th day(0 based indexing) and sell at 4th day.
4 Max profit = 99-12 =87
5
```

Solution : [Stock Buy Sell to Maximize Profit.](#)

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 :

```
1
2 int arr[]={14, 12, 70, 15, 95, 65, 22, 30};
3 Max Difference =95-12 = 83
4
```

Solution : [Maximum difference between two elements such that larger element appears after the smaller number.](#)

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.

Solution : [Search in a row wise and column wise sorted matrix.](#)

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 :

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

Solution : [Largest sum contiguous subarray.](#)

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 :

```
1
2 arr[]={14, 12, 70, 15, 99, 65, 21, 90};
3 X=97.
4
```

5 Sum found between index 1 to 3

6 Elements are 12, 17 and 15

Solution : [Find the Contiguous Subarray with Sum to a Given Value in an array.](#)

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 :

1

2 `String[] strArr={"java2blog","javaworld","javabean","javatemp"};`

3 So Longest common prefix in above **String array** will be "java" as all above **string** starts with "java".

4

Solution : [Longest Common Prefix in an array of Strings in 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 :

1

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

3 Output:

4 [

5 [3],

6 [1],

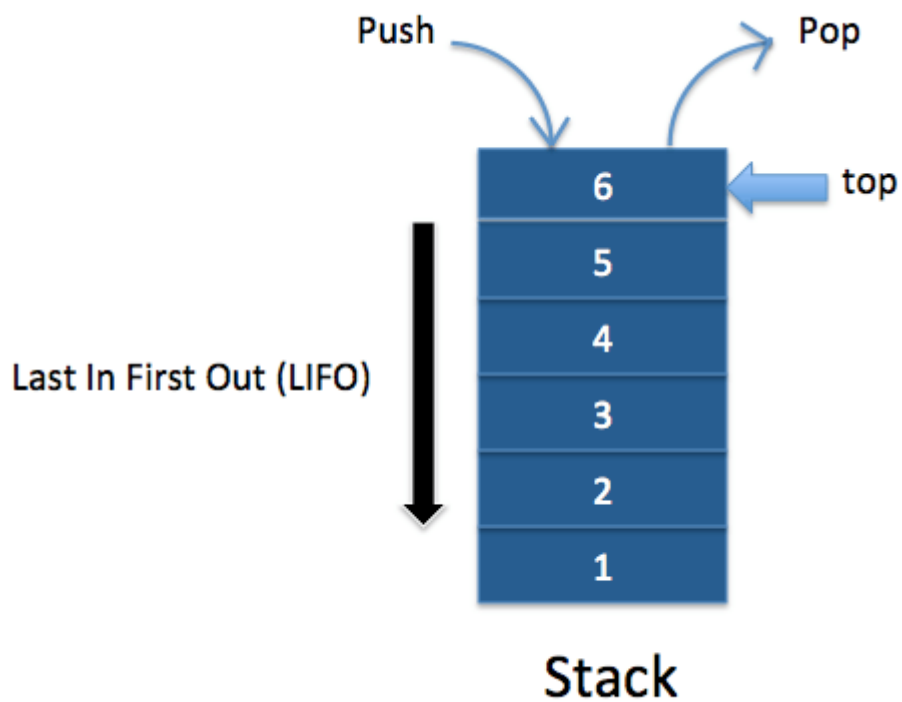
7 [2],

8


```
9 [1,2,3],  
10 [1,3],  
11 [2,3],  
12 [1,2],  
13 [],  
14 ]
```

Solution : [Find all subsets of set in java.](#)

Stack



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).

Solution : [Java Program to implement stack using array.](#)

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).

Solution : [Java Program to implement stack using Linked List](#)

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).

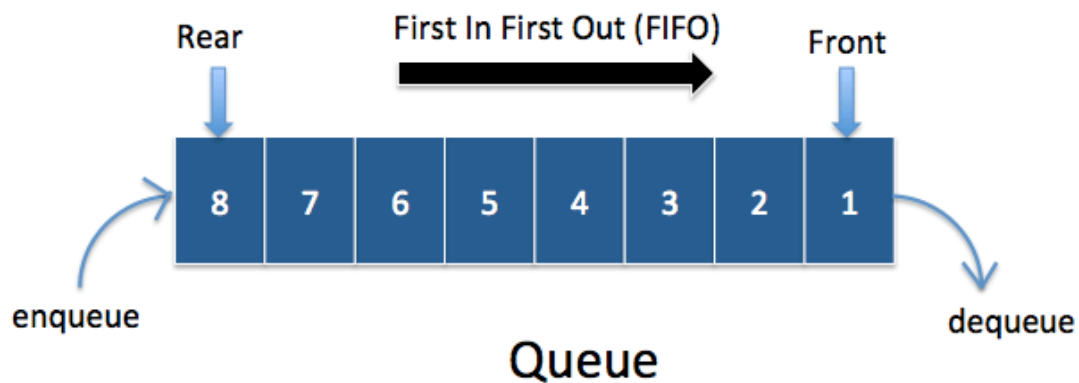
Solution : [Java Program to implement stack using two queues](#)

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,

Solution : [Sort a stack using another stack.](#)

Queue



Question 45: Implement Queue using Array in java.

You need to use array to implement queue.

Solution : [Implement Queue using Array in java](#)

Question 46: Implement a stack using two queues .

You need to use Linked list to implement queue.

Solution : [Java Program to implement queue using linked list](#)

Linked List

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.



Solution : [Java program to implement singly linked list in java.](#)

Question 48: How to reverse linked list in java.

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

Solution : [Java program to reverse linked list in java.](#)

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.



Solution : [Java program to find middle element of linked list.](#)

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.

Solution : [How to find nth element from end 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.

Solution : [How to detect loop in linked list.](#)

[How to find start node of loop in linked list.](#)

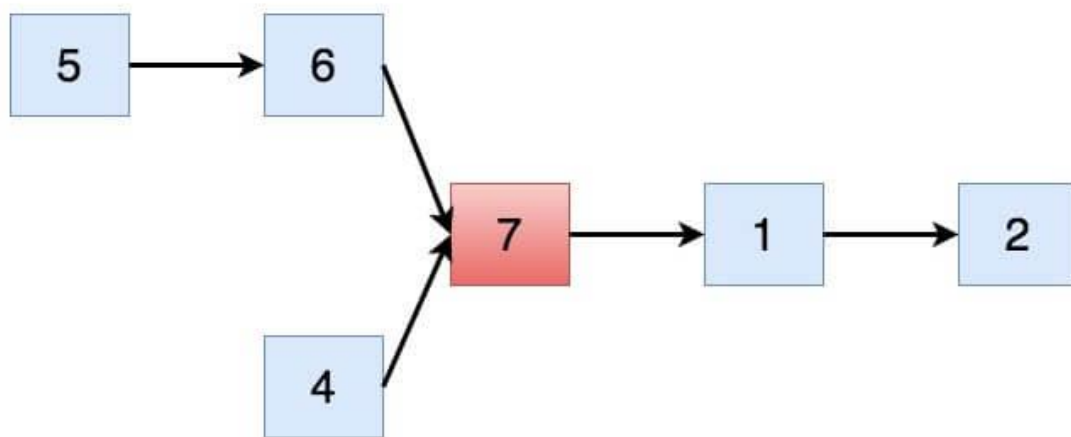
Question 52: How to check if linked list is palindrome or not?

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.

Solution : [Java program to check if linked list is palindrome.](#)

Question 53 : Find intersection of two linked lists?

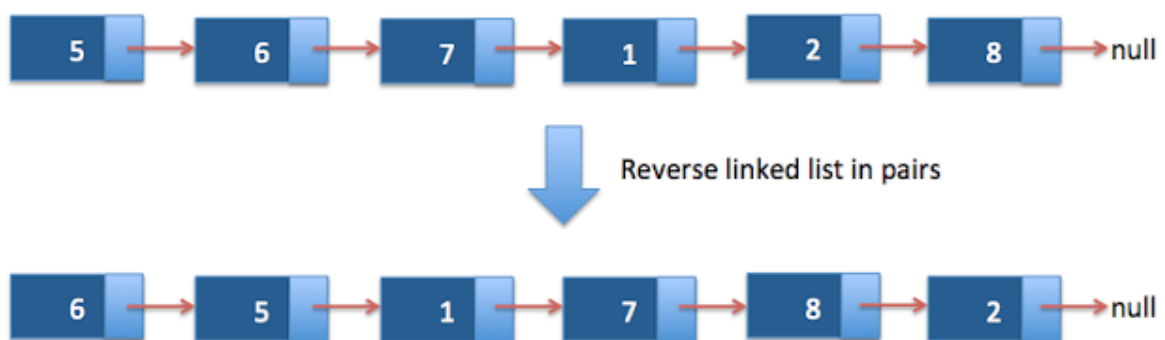
Given two [singly linked lists](#), find if two linked lists intersect. If they intersect, find intersection point.



Solution : [Intersection of two linked list](#)

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

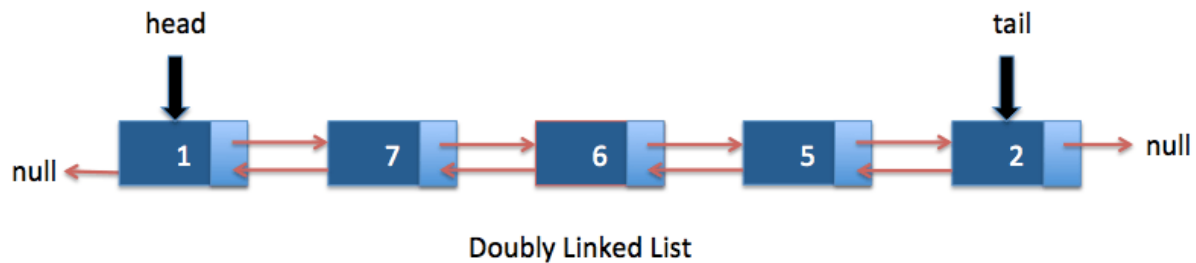
You need to write a java program to reverse linked list in pairs.



Solution : [Java program to reverse linked list in pair.](#)

Question 55 : Implement Doubly linked list in java?

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



Solution : [Doubly Linked List in java](#)

Binary Tree

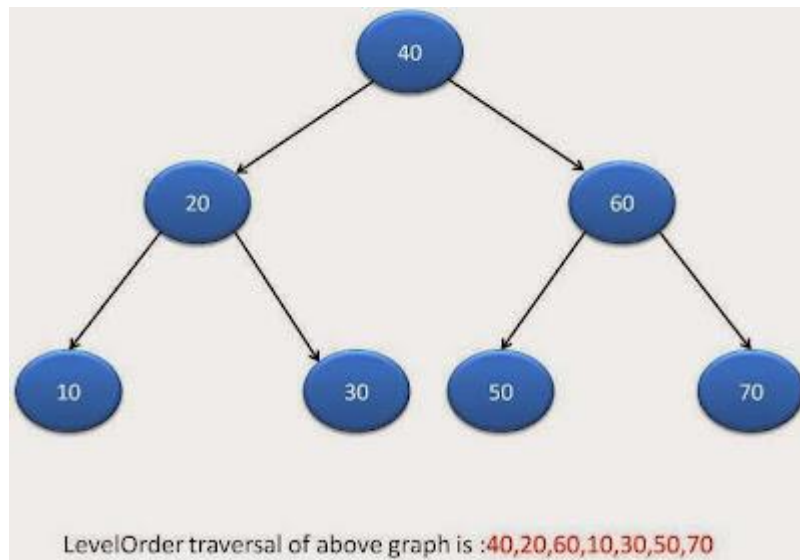
Question 56 : How can you traverse binary tree?

There are three ways to traverse binary tree.

- [PreOrder](#)
 - [InOrder](#)
 - [PostOrder](#).
-

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

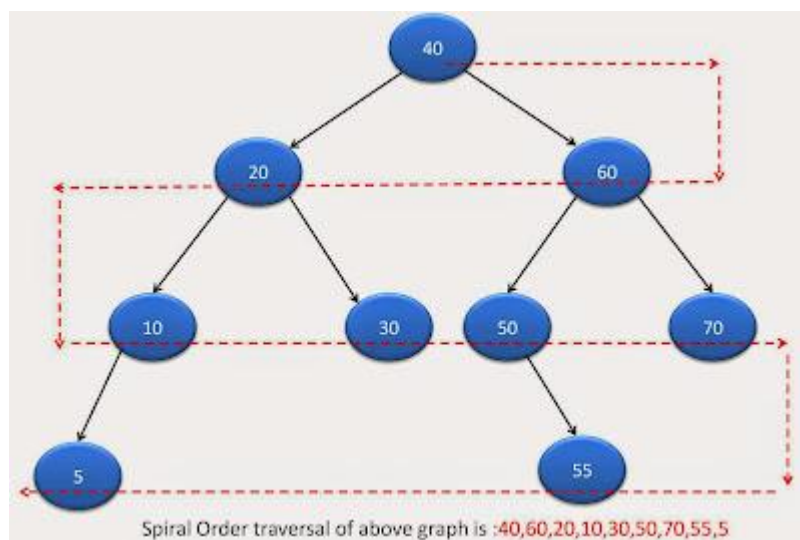
You need to write java program to do level order traversal of binary tree.
You can use queue data structure to do level order traversal.



Solution : [Level order traversal of binary tree.](#)

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

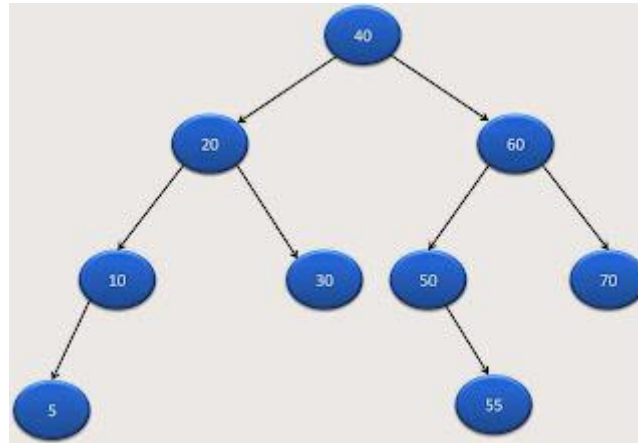
You need to write java program to do spiral level order traversal of binary tree



Solution : [Spiral order or zigzag 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.



Leaf nodes for above binary tree will be 5 , 30 , 55 ,70

Solution : [Print leaf nodes of binary tree.](#)

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

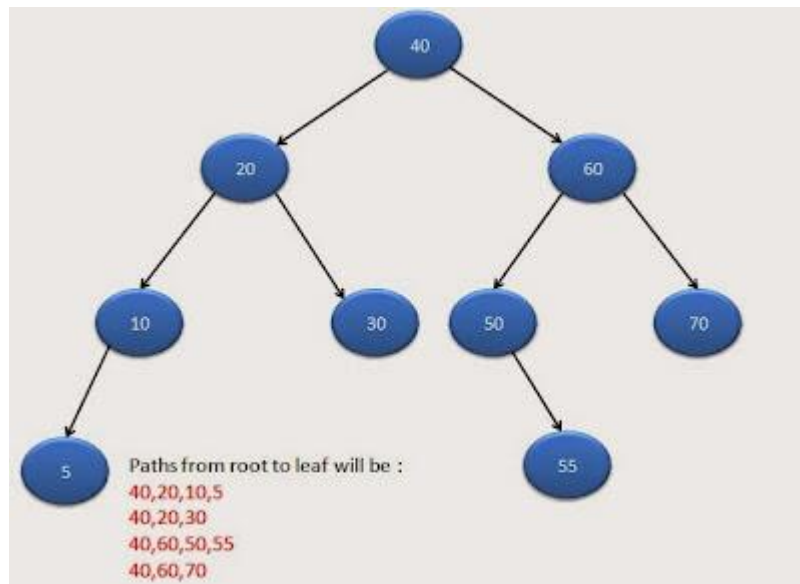
You need to write java program to count leaf nodes of binary tree.

Count of Leaf nodes for binary tree used in Question 15 are **5**.

Solution : [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.



Solution : [Print all paths from root to leaf in binary tree.](#)

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.

Solution: [Find level of node in binary tree.](#)

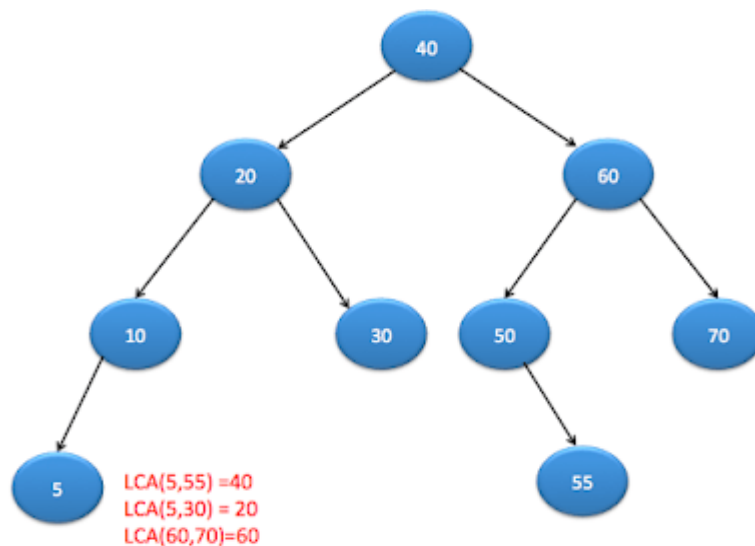
Question 63 : How to find maximum element in binary tree.

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

Solution : [Find maximum element in binary tree.](#)

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

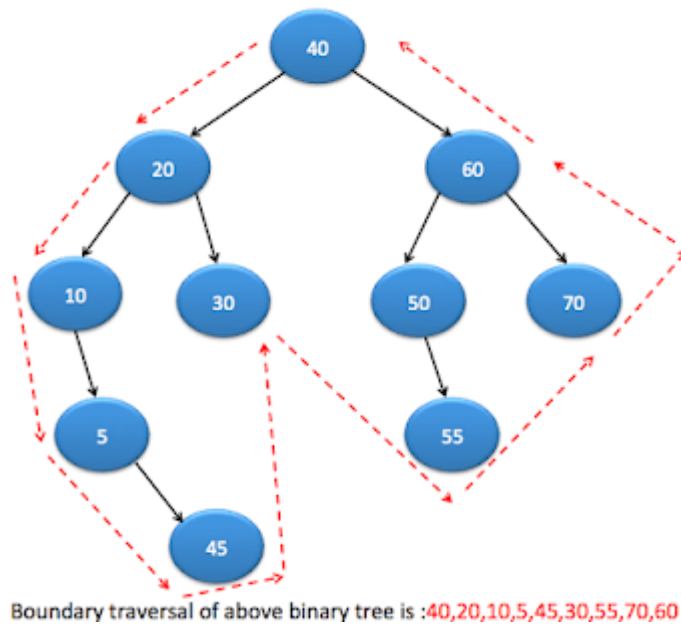
You need to write a program to find LCA in binary tree.



Solution: [Program to find LCA in binary tree.](#)

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

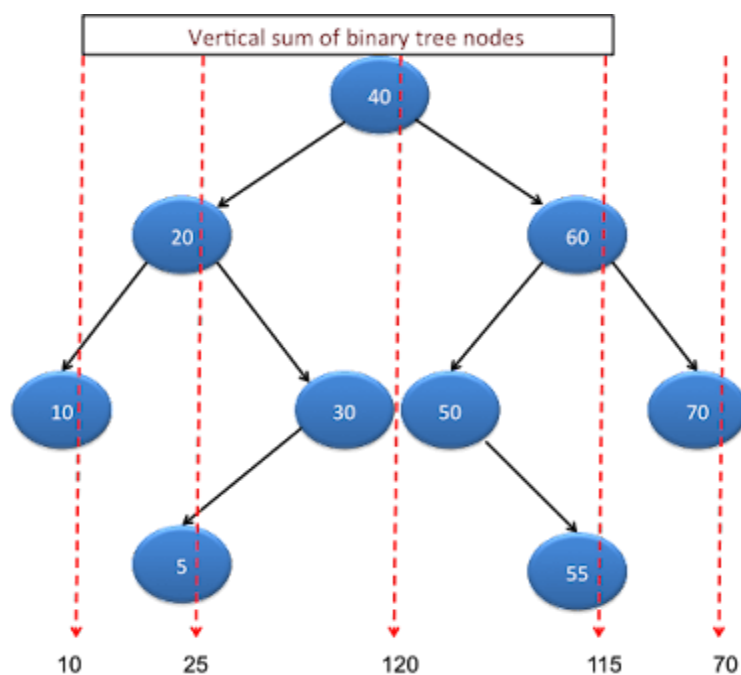
Write a java program to do boundary traversal of binary tree as shown in below image.



Solution : [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.



Solution : [How to print vertical sum of binary tree.](#)

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.

Solution : [Count subtrees with Sum equal to target in binary tree](#).

Binary Search tree

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.

Solution : [Insert node in binary search tree](#)

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

Solution : [Delete node in binary search tree](#)

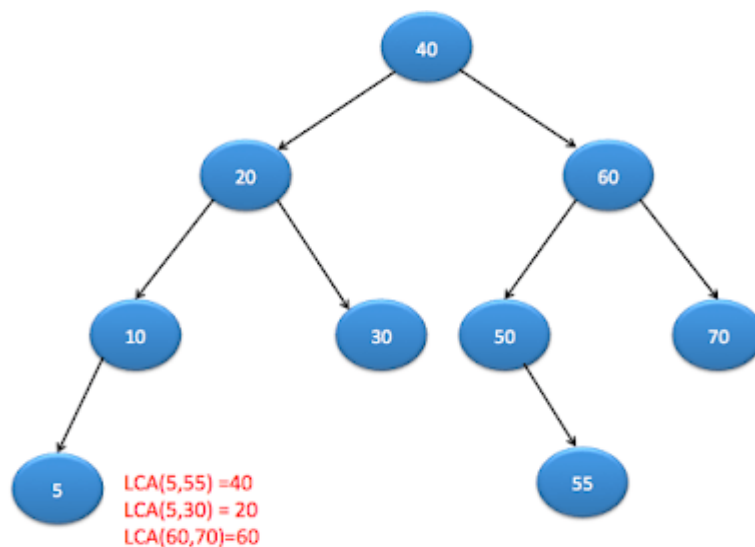
Question 71 : How can you find minimum and maximum elements in binary search tree?

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

[Minimum and maximum elements in binary search tree.](#)

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.



Solution: [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.

Solution: [Inorder Successor in a Binary Search Tree](#)

Question 74 : Convert sorted array to balanced BST

Solution: [Convert sorted sorted array to balanced BST](#)

Question 75 : Convert sorted Linked List to balanced BST

Solution: [Convert sorted Linked List to balanced BST](#)

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

Solution: [Check if a binary tree is binary search tree or not in java](#)

Sorting

Question 77 : Write an algorithm to implement bubble sort?

Solution : [Bubble sort in java](#)

Question 78 : Write an algorithm to implement insertion sort?

Solution : [Insertion sort in java](#)

Question 79 : Write an algorithm to implement selection sort?

Solution : [Selection sort in java](#)

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

Solution : [Merge sort in java](#)

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

Solution : [implement Heap sort in java](#)

Question 82 : Implement quick sort in java?

Solution : [implement Quick sort in java](#)

Question 83 : Implement shell sort in java?

Solution : [implement Shell sort in java](#)

Question 84 : Implement Counting sort in java?

Solution : [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?

Solution : [Binary search algorithm in java](#)

Graph

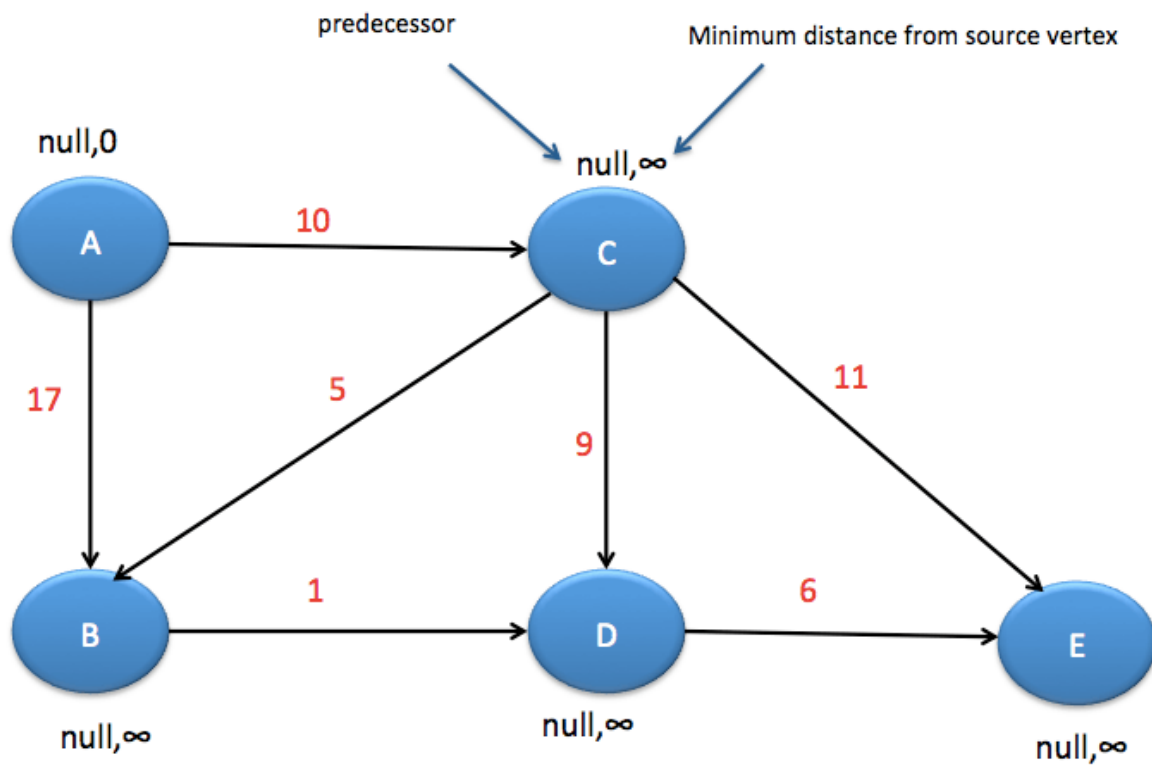
Question 86 : Write algorithm to do depth first search in a graph.

Solution : [Depth first search in java](#)

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

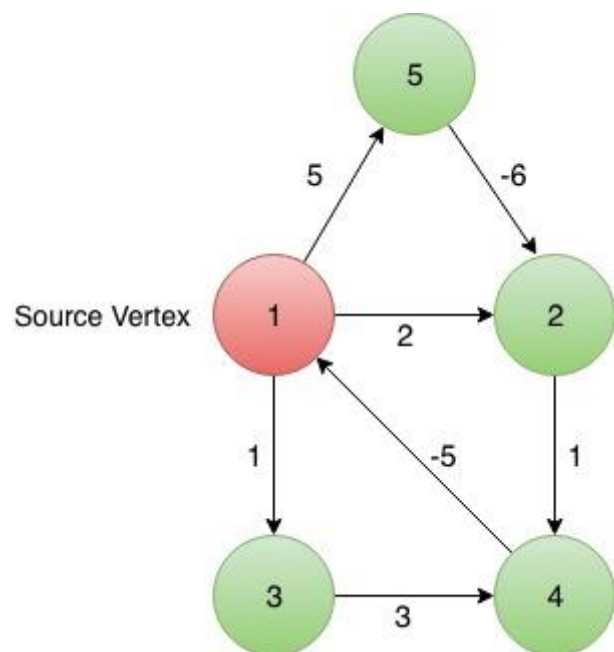
Solution : [breadth first search in java](#)

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



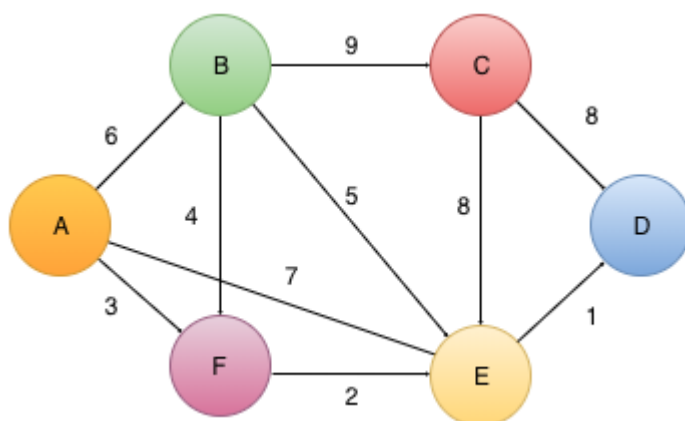
Solution : [Dijkstra's algorithm in java](#)

Question 89 : Explain Bellman Ford algorithm to find shortest distance



Solution : [Bellman ford algorithm in java](#)

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



Solution : [Kruskal's algorithm](#)

Dynamic Programming

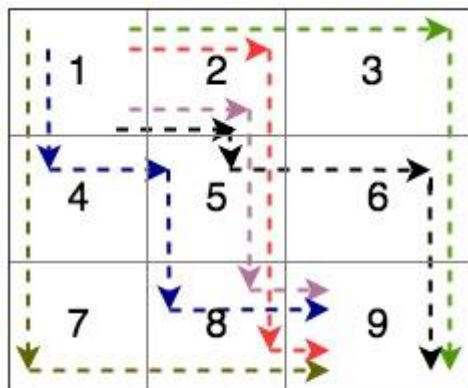
Question 91 : Given two String, find longest common substring.

Solution: [Longest common substring in java](#).

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

Solution: [Longest common subsequence in java](#)

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

Solution: [Count all paths in matrix](#)

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.

Solution: [Edit distance problem in java](#).

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.

Solution: [Coin change problem in java](#)

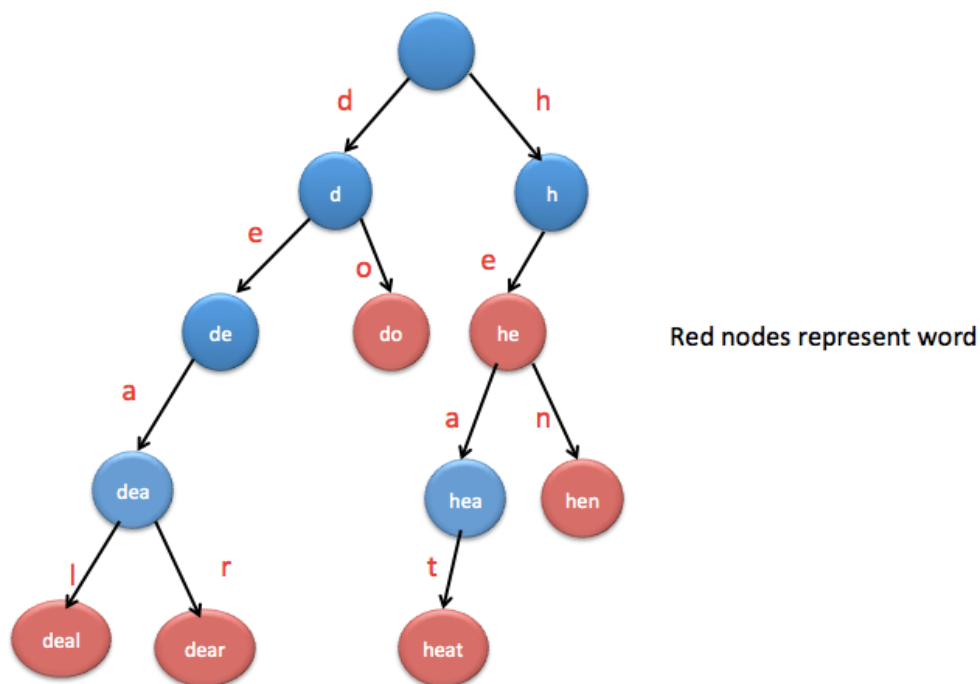
Solution: [Minimum number of jumps to reach last index](#).

Miscellaneous

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

Solution : [How to calculate Complexity of algorithm](#)

Question 98 : Implement trie data structure in java.



Trie Data structure

Solution : [Implement trie data structure in java.](#)

Question 99 : Count Factorial Trailing Zeroes in java.

Solution : [Count Factorial Trailing Zeroes in java](#)

Question 100 : Largest Rectangular Area in a Histogram.

Solution : [Count Largest Rectangular Area in a Histogram](#)

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

Solution : [check for balanced parentheses in an expression in java.](#)

Question 102 : What is Memoization.

Solution :

Memoization ensures that method does not execute more than once for same inputs by storing the results in the data structure(Usually Hashtable or [HashMap](#) or Array).

[Memoization example in java](#)

This is all about java coding interview questions. Please do comment if you want to add any new questions to above list.

You may also like: