***Data Structures And Linked Lists***

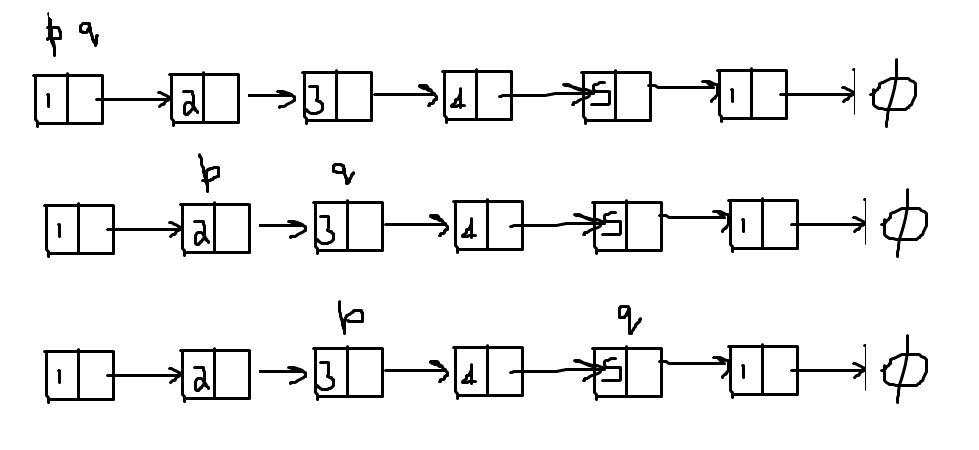
1.

**Find the middle of a given linked list in efficient time.**

Given a singly linked list, find middle of the linked list. For example, if given linked list is 1->2->3->4->5 then output should be 3.

If there are even nodes, then there would be two middle nodes, we need to print second middle element. For example, if given linked list is 1->2->3->4->5->6 then output should be 4.

🡺  
Traverse linked list using two pointers. Move one pointer by one and other pointer by two. When the fast pointer reaches end slow pointer will reach middle of the linked list because it would have travelled just the half of what fast pointer has travelled. Since fast pointer has traversed the full list , slow pointer will be at middle at that moment.



(aka hare and tortoise method.)

Similarly if we want the 1/3rd of the list , move the second pointer three steps.

2.

**Flattening a Linked List**

Given a linked list where every node represents a linked list and contains two pointers of its type:  
(i) Pointer to next node in the main list (we call it ‘right’ pointer in below code)  
(ii) Pointer to a linked list where this node is head (we call it ‘down’ pointer in below code).  
All linked lists are sorted. See the following example

5 -> 10 -> 19 -> 28

| | | |

V V V V

7 20 22 35

| | |

V V V

8 50 40

| |

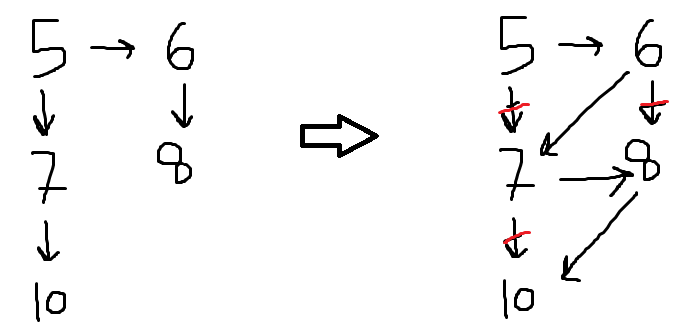
V V

30 45

Write a function flatten() to flatten the lists into a single linked list. The flattened linked list should also be sorted. For example, for the above input list, output list should be 5->7->8->10->19->20->22->28->30->35->40->45->50.

🡺

We need to make a single list out of the above structure. If we merge two lists at a time, we will eventually get a single list. Since the lists are already sorted, using merge process of merge sort makes sense.



This way we can merge all lists, taking two at once.

3.

**Find sum with Zero**

Given some resources in the form of linked list you have to cancel out all the resources whose sum up to 0(Zero) and return the remaining list.

Note : Only consecutive elements are required for zero sum  
  
Example 6 -6 8 4 -12 9 8 -8  
the above example lists which gets cancelled :   
6 -6   
8 4 -12   
8 -8   
o/p : 9   
case 3 : 4 6 -10 8 9 10 -19 10 -18 20 25   
O/P : 20 25

🡺

We need the consecutive groups with zero sum.

If we add up all the elements, we will get the sum of remaining elements. Now if we can find out which elements are having this sum , we can simply print those elements.

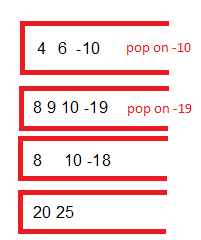
For example 6 -6 8 4 -12 9 8 -8

Sum is 9 , and for this example we directly have 9 as the group having sum = 9

For 4 6 -10 8 9 10 -19 10 -18 20 25

Sum is 45 , and the group having this sum is 20 25

🡺We can also use a stack. We can push positive numbers and pop an equivalent sum of negative numbers. This makes sense because we will get only consecutive numbers in input that sum up to 0.



In the same problem if the ‘zero-sum-elements’ were not consecutive then it would have become NP Complete subset sum problem which would take exponential time.

4.

**Remove duplicate element from sorted Linked List**

Given a linked list of N nodes. The task is to remove duplicates from the given list (if exists).

For example if the linked list is 11->11->11->21->43->43->60, then linked list should be converted to 11->21->43->60.

Note: Try not to use extra space. Expected time complexity is O(N).

Input:  
First line of input contains number of testcases T. For each testcase, first line of input contains length of linked list and next line contains the linked list data.

Output:  
For each testcase, there will be a single line of output which contains linked list with no duplicates.

Hint:

Traverse each node(P) and put a pointer(Q) to its next node and move that pointer(Q) forward while P.data==Q.data. On termination, P.next=Q and move P forward.

5.

**Delete Middle of Linked List**

Given a singly linked list, delete middle of the linked list. For example, if given linked list is 1->2->3->4->5 then linked list should be modified to 1->2->4->5

If there are even nodes, then there would be two middle nodes, we need to delete the second middle element. For example, if given linked list is 1->2->3->4->5->6 then it should be modified to 1->2->3->5->6.

If the input linked list is NULL, then it should remain NULL.

If the input linked list has 1 node, then this node should be deleted and new head should be returned.

Hint:

Use two pointers, P and Q. P should move one step at a time and Q should move two steps.

6.

**Add 1 to a number represented as linked list**

Number is represented in linked list such that each digit corresponds to a node in linked list. Add 1 to it. For example 1999 is represented as (1-> 9-> 9 -> 9) and adding 1 to it should change it to (2->0->0->0)

Below are the steps :

Reverse given linked list. For example, 1-> 9-> 9 -> 9 is converted to 9-> 9 -> 9 ->1.

Start traversing linked list from leftmost node and add 1 to it. If there is a carry, move to the next node. Keep moving to the next node while there is a carry.

Reverse modified linked list and return head.

7.

**Reverse a Linked List in groups of given size.**

Given a linked list of size N. The task is to reverse every k nodes (where k is an input to the function) in the linked list.

Eg k = 3

list is :: 4=5=9=6=4=8=7=4=1=5=5=4

required list:: 9=5=4=8=4=6=1=4=7=4=5=5

Hint:

1) Reverse the first sub-list of size k. While reversing keep track of the next node and previous node. Let the pointer to the next node be next and pointer to the previous node be prev.

2) head->next = reverse(next, k) /\* Recursively call for rest of the list and link the two sub-lists \*/  
3) return prev /\* prev becomes the new head of the list

8.

Detect Loop in linked list

Given a linked list of N nodes. The task is to check if the the linked list has a loop. Linked list can contain self loop.

Input:  
First line of input contains number of testcases T. For each testcase, first line of input contains length of linked list and next line contains the data of linked list.

Output:  
For each testcase, print "True", if linked list contains loop, else print "False".

User Task:  
The task is to complete the function detectloop() which contains reference to the head as only argument. This function should return 1 if linked list contains loop, else return 0.

Constraints:  
1 <= T <= 50  
1 <= N <= 300

Example:  
Input:  
2  
3  
1 3 4  
2  
4  
1 8 3 4  
0

Output:  
True  
False

Explaination:  
Testcase 1: In above test case N = 3. The linked list with nodes N = 3 is given. Then value of x=2 is given which means last node is connected with xth node of linked list. Therefore, there exists a loop.

Testcase 2: For N = 4  
x = 0 means then lastNode->next = NULL, then the Linked list does not contains any loop.

Hint:  
1. Use hashing to detect the first node, the \*next of which has already occurred once.

OR

2 . Use Floyd’s Loop detection algo.

9.

**Remove loop in Linked List**

You are given a linked list of N nodes. You need to remove the loop if present.

Input Format:  
First line of input contains number of test cases T. T test cases follow. For each test case, first line of input contains length N of the linked list and next line contains N data of the linked list.  The third line contains the position of the node (from head) to which the last node will get connected. If it is 0 then there is no loop.

Output:  
For each test case, in a new line, 1 will be printed if loop is removed(correct answer) else 0 will be printed (for wrong answer).

Constraints:  
1 <= T <= 50  
1 <= N <= 300

Example:  
Input:  
2  
3  
1 3 4  
2  
4  
1 8 3 4  
0

Output:  
1  
1

Explanation:  
Testcase 1: In the first test case N = 3  
The linked list with nodes N = 3 is given. here x=2 which means last node is connected with xth node of linked list. Therefore, there exists a loop.

Testcase 2: N = 4 and x = 0, which means lastNode->next = NULL, thus the Linked list does not contains any loop.

Hint:  
Detect the last node of the loop by hashing or floyd’s algo(By adding length of loop to the head) and make its next NULL.

10.

**Nth node from end of linked list**

Given a linked list of length L. The task is to find the Nth node from the end of the linked list.  It is needed to complete a method that takes two argument, head of linked list and an integer N. There are multiple test cases. For each test case, this method will be called individually.

Input:  
First line of input contains number of testcase T. For each testcase, first line of input contains number of nodes in the linked list and N. Next line contains nodes of linked list.

Output:  
For each test case, output the data of node which is at Nth distance from end.

Hint:

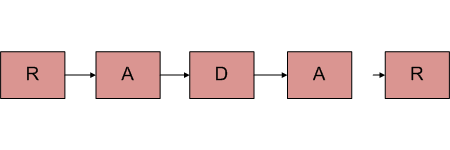
Take two pointers P and Q. Move P to the nth node.

Now move both P and Q while P!=NULL. Q will be reach to the nth node.

11.

**Function to check if a singly linked list is palindrome**

Given a singly linked list of characters, write a function that returns true if the given list is palindrome, else false.



Hint:  
Use hare and tortoise method to reach to the mid node. From there, reverse the rest of the list and then compare both parts.

12.

**Delete last occurrence of an item from linked list**

Given a liked list and a key to be deleted. Delete last occurrence of key from linked. The list may have duplicates.

Examples:

Input: 1->2->3->5->2->10, key = 2

Output: 1->2->3->5->10

Hint:

Traverse the list and maintain a pointer P to the node having data=k

Whenever such element is encountered, update P.

After traversing, print P->data.

13.

Rotate a Linked List

Given a singly linked list, rotate the linked list counter-clockwise by k nodes. Where k is a given positive integer. For example, if the given linked list is 10->20->30->40->50->60 and k is 4, the list should be modified to 50->60->10->20->30->40. Assume that k is smaller than the count of nodes in linked list.

Hint:

Kth-node ->next = NULL

Last->next = Head

14.

Merge a linked list into another linked list at alternate positions

Given two linked lists, insert nodes of second list into first list at alternate positions of first list.  
For example, if first list is 5->7->17->13->11 and second is 12->10->2->4->6, the first list should become 5->12->7->10->17->2->13->4->11->6 and second list should become empty. The nodes of second list should only be inserted when there are positions available. For example, if the first list is 1->2->3 and second list is 4->5->6->7->8, then first list should become 1->4->2->5->3->6 and second list to 7->8.

Use of extra space is not allowed (Not allowed to create additional nodes), i.e., insertion must be done in-place. Expected time complexity is O(n) where n is number of nodes in first list.

Hint :

r = head2

s = null

for each node t in list1:

s=r->next;

r->next = t->next;

t->next = r;

t=r->next;

r=s;

# 15.

# Convert singly linked list into circular linked list

Given a singly linked list, we have to convert it into circular linked list. For example, we have been given a singly linked list with four nodes and we want to convert this singly linked list into circular linked list.

Approach: The idea is to traverse the singly linked list and check if the node is the last node or not. If the node is the last node i.e pointing to NULL then make it point to the starting node i.e head node. Below is the implementation of this approach.

**16. Implement two stacks in an array**

**Create a data structure twoStacks that represents two stacks. Implementation of twoStacks should use only one array, i.e., both stacks should use the same array for storing elements. Following functions must be supported by twoStacks.**

*Start with two pointers, i=0 and j=n-1 (if n is size of the array)*

*the stack will grow towards the center of the array.*

**17.Implement Stack using Queues**

We can use 2 queues. For push operation, insert the element in the empty queue, one by one put all the elements of the other queue by dequeue() operation. For pop opeartion use dequeue() operation on the non empty queue.

**18.Design a stack with operations on middle element**

**How to implement a stack which will support following operations in** O(1) time complexity**?  
1) push() which adds an element to the top of stack.  
2) pop() which removes an element from top of stack.  
3) findMiddle() which will return middle element of the stack.  
4) deleteMiddle() which will delete the middle element.**

*Make a Doubly linked list to implement this stack. Use hare and tortoise method in order to maintain the middle point of the stack(after two push or two pops, move the middle pointer back/forth.)*

1. How to efficiently implement k stacks in a single array?

We can divide the array in n//k slots. Other approach can take two extra arrays where we can store the top of each stack in first array while storing the next-to-top pointer in second array(just what we do in a linked list).

**20.**Question:**Design a Data Structure SpecialStack that supports all the stack operations like push(), pop(), isEmpty(), isFull() and an additional operation getMin() which should return minimum element from the SpecialStack. All these operations of SpecialStack must be O(1).**

**Ans:**

We can use two stacks for that, one for implementing the stack of elements and other for maintaining the min-element for a certain level.

One more approach is::  
**Push(x)** : Inserts x at the top of stack.

* If stack is empty, insert x into the stack and make minEle equal to x.
* If stack is not empty, compare x with minEle. Two cases arise:
  + If x is greater than or equal to minEle, simply insert x.
  + If x is less than minEle, insert (2\*x – minEle) into the stack and make minEle equal to x. For example, let previous minEle was 3. Now we want to insert 2. We update minEle as 2 and insert 2\*2 – 3 = 1 into the stack.

**Pop() :**Removes an element from top of stack.

* Remove element from top. Let the removed element be y. Two cases arise:
  + If y is greater than or equal to minEle, the minimum element in the stack is still minEle.
  + If y is less than minEle, the minimum element now becomes (2\*minEle – y), so update (minEle = 2\*minEle – y). This is where we retrieve previous minimum from current minimum and its value in stack. For example, let the element to be removed be 1 and minEle be 2. We remove 1 and update minEle as 2\*2 – 1 = 3.

(from geeksforgeeks.org)

1. **Implement stack using *single* queue.**

Use n rotations to pop any element.

Eg at a certain point the stack(using queue) looks like:

2-5-8-9 (front = 9 , last element inserted was 2)

pop() will work in this manner(for n-1 iterations where n = size of queue at any point) :

dequeue 9 and enqueue it 9-2-5-8

dequeue 8 and enqueue it 8-9-2-5

eventually, 2 will be at front , then dequeue it and return it.

22. **Algorithm for Prefix to Infix**:

* Read the Prefix expression in reverse order (from right to left)
* If the symbol is an operand, then push it onto the Stack
* If the symbol is an operator, then pop two operands from the Stack  
  Create a string by concatenating the two operands and the operator between them.  
  **string = (operand1 + operator + operand2)**  
  And push the resultant string back to Stack
* Repeat the above steps until end of Prefix expression.

23.**Algorithm for Prefix to Postfix**:

* Read the Prefix expression in reverse order (from right to left)
* If the symbol is an operand, then push it onto the Stack
* If the symbol is an operator, then pop two operands from the Stack  
  Create a string by concatenating the two operands and the operator after them.  
  **string = operand1 + operand2 + operator**  
  And push the resultant string back to Stack
* Repeat the above steps until end of Prefix expression.

24.**postfix to prefix**

same as 23 but read the string from left to right.

25.The Celebrity Problem

In a party of N people, only one person is known to everyone. Such a person may be present in the party, if yes, (s)he doesn’t know anyone in the party. We can only ask questions like “does A know B? “. Find the stranger (celebrity) in minimum number of questions.

We can describe the problem input as an array of numbers/characters representing persons in the party. We also have a hypothetical function HaveAcquaintance(A, B) which returns true if A knows B, false otherwise. How can we solve the problem.

We measure the complexity in terms of calls made to HaveAcquaintance().

Method 1 (Graph)

We can model the solution using graphs. Initialize indegree and outdegree of every vertex as 0. If A knows B, draw a directed edge from A to B, increase indegree of B and outdegree of A by 1. Construct all possible edges of the graph for every possible pair [i, j]. We have NC2 pairs. If celebrity is present in the party, we will have one sink node in the graph with outdegree of zero, and indegree of N-1. We can find the sink node in (N) time, but the overall complexity is O(N2) as we need to construct the graph first.

Method 2 (Recursion)

We can decompose the problem into combination of smaller instances. Say, if we know celebrity of N-1 persons, can we extend the solution to N? We have two possibilities, Celebrity(N-1) may know N, or N already knew Celebrity(N-1). In the former case, N will be celebrity if N doesn’t know anyone else. In the later case we need to check that Celebrity(N-1) doesn’t know N.

Solve the problem of smaller instance during divide step. On the way back, we find the celebrity (if present) from the smaller instance. During combine stage, check whether the returned celebrity is known to everyone and he doesn’t know anyone. The recurrence of the recursive decomposition is,

T(N) = T(N-1) + O(N)

T(N) = O(N2). You may try writing pseudo code to check your recursion skills.

Method 3 (Using Stack)  
The graph construction takes O(N2) time, it is similar to brute force search. In case of recursion, we reduce the problem instance by not more than one, and also combine step may examine M-1 persons (M – instance size).

We have following observation based on elimination technique (Refer Polya’s How to Solve It book).

* If A knows B, then A can’t be celebrity. Discard A, and B may be celebrity.
* If A doesn’t know B, then B can’t be celebrity. Discard B, and A may be celebrity.
* Repeat above two steps till we left with only one person.
* Ensure the remained person is celebrity. (Why do we need this step?)

We can use stack to verity celebrity.

1. Push all the celebrities into a stack.
2. Pop off top two persons from the stack, discard one person based on return status of HaveAcquaintance(A, B).
3. Push the remained person onto stack.
4. Repeat step 2 and 3 until only one person remains in the stack.
5. Check the remained person in stack doesn’t have acquaintance with anyone else.

We will discard N elements utmost (Why?). If the celebrity is present in the party, we will call HaveAcquaintance() 3(N-1) times. Here is code using stack.

26. Rotate the given square matrix in clockwise direction.

==> take transpose of the matrix and find its mirror image(Y axis)

27. Delete the alternate nodes of a given linked list.

==> Simply use two pointers and move one pointer two steps at a time keeping it next node of the other pointer.

28. What are the nodes on which rotation is performed in a balanced BST.

Ans: The first unbalanced ancestor of the newly inserted node.

29. Given a sorted array, say 1 2 3 4 5 6, we perform some rotations on it. The array becomes 3 4 5 6 1 2. Find the number of rotations.

==> position of first elements in sorted array

30. Given an array, find f(x) for each x in array where f(x) = product\_of\_array/x. Division operator is not allowed.

==>Make two cumulative arrays that store the prefix and suffix product respectively. For each x, print prefix(i-1)\*suffix(i+1).

31.[How is an Array different from Linked List?](https://www.geeksforgeeks.org/linked-list-vs-array/)

* The size of the arrays is fixed, Linked Lists are Dynamic in size.
* Inserting and deleting a new element in an array of elements is expensive, Whereas both insertion and deletion can easily be done in Linked Lists.
* Random access is not allowed in Linked Listed.
* Extra memory space for a pointer is required with each element of the Linked list.
* Arrays have better cache locality that can make a pretty big difference in performance.

32. What is Stack and where it can be used?

Stack is a linear data structure which the order LIFO(Last In First Out) or FILO(First In Last Out) for accessing elements. Basic operations of stack are : Push, Pop , Peek

Applications of Stack:

1. [Infix to Postfix Conversion using Stack](http://geeksquiz.com/stack-set-2-infix-to-postfix/)
2. [Evaluation of Postfix Expression](http://geeksquiz.com/stack-set-4-evaluation-postfix-expression/)
3. [Reverse a String using Stack](http://geeksquiz.com/stack-set-3-reverse-string-using-stack/)
4. [Implement two stacks in an array](https://www.geeksforgeeks.org/implement-two-stacks-in-an-array/)
5. [Check for balanced parentheses in an expression](https://www.geeksforgeeks.org/check-for-balanced-parentheses-in-an-expression/)

33. Difference between RB tree and AVL Tree.

RB tree is having less rotations for insertion/deletions so it is preferred when we have frequent insertion/deletion.

AVL tree is used for frequent search operations.

34. Inorder traversal without recursion.

(Use stack to store the return state or use iterative approach)

35. find all possible binary trees from given inorder traversal

From each of the element in given array, split the array in two parts and repeat this step recursively

Eg 1,2,3,4,5

First make a tree for left = null , root = 1 , right = 2,3,4,5

Then for left = null, recur and for right = 2,3,4,5 ,recur

Similarily for left = 1, root = 2 and right = 3,4,5 , repeat above process

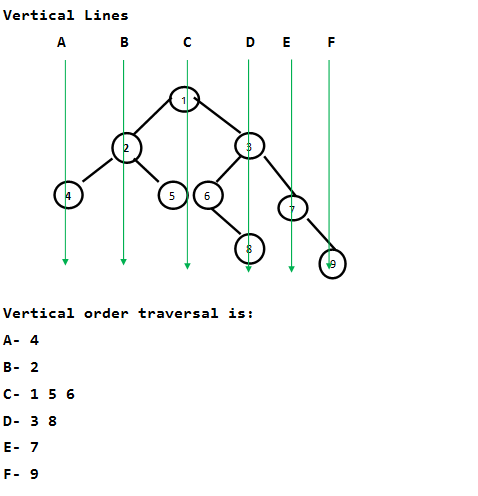
36. find post order traversal from given preorder and inorder traversal

Use the property of preorder that first node is always root of that subtree and from that node, split the inorder list into two parts, left one will be left subtree and right one will be right.

37. [Replace each node in binary tree with the sum of its inorder predecessor and successor](https://www.geeksforgeeks.org/replace-node-binary-tree-sum-inorder-predecessor-successor/)

Store inorder traversal in an array and then again traverse the tree while replacing each node by arr[i-1]+arr[i+1]

38.find vertical traversal of a tree..



Put root.lvl = 0

While traversing the tree if you visit the left node, pass lvl = current\_lvl-1 and current\_lvl+1 for right node.

The nodes with similar lvl value will be on same levels.

(Use map or dictionary of lists)

39. Find maximum vertical sum in a tree

Use the similar approach as in Q 38 , just add the values of nodes of each level and return the max of all of them.

40.

Given a Binary Tree where each node has following structure, write a function to populate next pointer for all nodes. The next pointer for every node should be set to point to inorder successor.

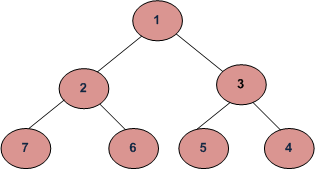
|  |
| --- |
| struct node  {    int data;    struct node\* left;    struct node\* right;    struct node\* next;  } |

Ans: initialize a global pointer(say ptr) to left most node of tree.

Traverse the tree (say root is that pointer)After each visit, set ptr->next = root , ptr = root

41.

Write a function to print spiral order traversal of a tree. For below tree, function should print 1, 2, 3, 4, 5, 6, 7.



Ans: Traverse the tree using bfs

At even levels , print the nodes ,

At odd levels , store the nodes in a stack and print them at the end (while popping them)

In this way we can reverse the tree level wise

42. What is augumented data structure?

Basically a modified standard data structure. Eg in Q40, the node is modified. Other examples are fenwick tree , min query trees etc.

# 43. Level order traversal with direction change after every two levels

Given a binary tree, print the level order traversal in such a way that first two levels are printed from left to right, next two levels are printed from right to left, then next two from left to right and so on. So, the problem is to reverse the direction of level order traversal of binary tree after every two levels.

Examples:

Input:

1

/ \

2 3

/ \ / \

4 5 6 7

/ \ / \ / \ / \

8 9 3 1 4 2 7 2

/ / \ \

16 17 18 19

Output:

1

2 3

7 6 5 4

2 7 2 4 1 3 9 8

16 17 18 19

In the above example, first two levels

are printed from left to right, next two

levels are printed from right to left,

and then last level is printed from

left to right.

Ans: use the similar approach as in previous question but replace the even odd conditions with the appropriate ones.

# 44. Level order traversal with direction change after every two levels

Given a binary tree, print the level order traversal in such a way that first two levels are printed from left to right, next two levels are printed from right to left, then next two from left to right and so on. So, the problem is to reverse the direction of level order traversal of binary tree after every two levels.

Examples:

Input:

1

/ \

2 3

/ \ / \

4 5 6 7

/ \ / \ / \ / \

8 9 3 1 4 2 7 2

/ / \ \

16 17 18 19

Output:

1

2 3

7 6 5 4

2 7 2 4 1 3 9 8

16 17 18 19

In the above example, first two levels

are printed from left to right, next two

levels are printed from right to left,

and then last level is printed from

left to right.

Ans: search the node using dfs and store the nodes in the path using an array or map

\_ie if the path to node x is a->b->c->x

Then store it level wise, note that the data in this map/array is pointer to that node

In the end, use two pointers i=0 and j = x.index() and swap the data of these pointers while moving them to middle

45. Given a [Perfect Binary Tree](http://xlinux.nist.gov/dads/HTML/perfectBinaryTree.html) like below:  
(click on image to get a clear view)

image(4)

Print the level order of nodes in following specific manner:

1 2 3 4 7 5 6 8 15 9 14 10 13 11 12 16 31 17 30 18 29 19 28 20 27 21 26 22 25 23 24

i.e. print nodes in level order but nodes should be from left and right side alternatively. Here 1st and 2nd levels are trivial.  
While 3rd level: 4(left), 7(right), 5(left), 6(right) are printed.  
While 4th level: 8(left), 15(right), 9(left), 14(right), .. are printed.  
While 5th level: 16(left), 31(right), 17(left), 30(right), .. are printed.

Ans: store each level in a vector or queue and then print them in this format.

**Approach 2:**  
The standard level order traversal idea will slightly change here. Instead of processing ONE node at a time, we will process TWO nodes at a time. And while pushing children into queue, the enqueue order will be: 1st node’s left child, 2nd node’s right child, 1st node’s right child and 2nd node’s left child.(space efficient)

46. [image(4)](https://cdncontribute.geeksforgeeks.org/wp-content/uploads/binary_tree-1.png)

16 31 17 30 18 29 19 28 20 27 21 26 22 25 23 24 8 15 9 14 10 13 11 12 4 7 5 6 2 3 1

The task is to print nodes in level order but nodes should be from left and right side alternatively and from bottom – up manner

Ans:

One approach can be similar to previous question but instead of printing it directly, store each level on a stack or use a recursive approach.

47. Given a [Perfect Binary Tree](http://en.wikipedia.org/wiki/Binary_tree#Types_of_binary_trees), reverse the alternate level nodes of the binary tree.

Given tree:

a

/ \

b c

/ \ / \

d e f g

/ \ / \ / \ / \

h i j k l m n o

Modified tree:

a

/ \

c b

/ \ / \

d e f g

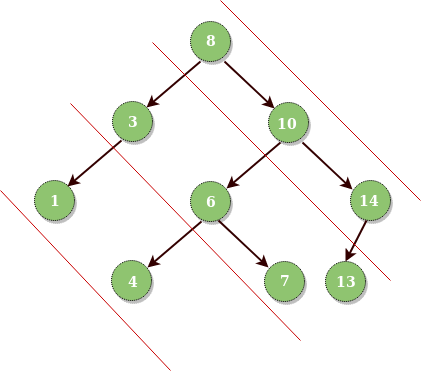
/ \ / \ / \ / \

o n m l k j i h

Ans: Use Q41 approach and store the level wise traversal. Traverse the tree again and replace the nodes from that array.

48. Consider lines of slope -1 passing between nodes. Given a Binary Tree, print all diagonal elements in a binary tree belonging to same line.

Input : Root of below tree



Output :

Diagonal Traversal of binary tree :

8 10 14

3 6 7 13

1 4

The idea is to use a queue to store only the left child of **current** node. After printing the data of current node make the current node to its right child, if present.  
A delimiter **NULL** is used to mark the starting of next diagonal.

Push root and null. Now as soon as a diagonal ends , you will encounter null in queue ie you should start a new diagonal from here. So keep pushing null after end of each diagonal.

49. Given a binary tree, print boundary nodes of the binary tree Anti-Clockwise starting from the root. For example, boundary traversal of the following tree is “20 8 4 10 14 25 22”

Ans:

Print the left arm(excluding the last node) and don’t forget to check the right child of nodes.

Then print all the leaves.(use any traversal)

Finally use recursion or stack to print the right arm in reverse.

50. Given a Binary Tree, find density of it by doing one traversal of it.

**Density of Binary Tree = Size / Height**

Ans: one method can be using bfs. Store the level of each node with them.

Initially size = 0 , root.lvl = 0

While pushing each child to queue , do size++ ; and set child.lvl = par.lvl+1;

In the end return size/last\_node.lvl

51. Count the number of Binary Tree possible for a given Preorder Sequence length n.

**Examples:**

Input : n = 1

Output : 1

Input : n = 2

Output : 2

Input : n = 3

Output : 5

Possible number of Binary Tree are:  
**BT(5)** = BT(0) \* BT(4) + BT(1) \* BT(3) + BT(2) \* BT(2) + BT(3) \* BT(1) + BT(4) \* BT(0)  
**BT(5)** = 1 \* 14 + 1 \* 5 + 2 \* 2 + 5 \* 1 + 14 \* 1 = 42

Hint: use dynamic programming to calculate the above relation.

This can also be done using [Catalan number](https://www.geeksforgeeks.org/program-nth-catalan-number/) Cn = (2n)!/(n+1)!\*n!

52. Given Linked List Representation of Complete Binary Tree, construct the Binary tree. A complete binary tree can be represented in an array in the following approach.

If root node is stored at index i, its left, and right children are stored at indices 2\*i+1, 2\*i+2 respectively.

Eg 10->20->30->40->50->60

Will form a tree:

10

/ \

20 30

/ \ / \

40 50 60 null

1.Make a queue = {}

2.Initialize levelcount = 1

3.Push levelcount elements in queue ie {10}

This is the root. Now level count \*= 2

Repeat 3 until we get null.

53. Given the preorder and postorder of a full binary tree, construct the tree.

Use preorder and postorder properties.

54. What is auxiliary space.

Extra space taken by an algorithm. In space complexity calculation we consider both auxiliary space and input size.

55. **What is Pseudo-polynomial?**  
An algorithm whose worst case time complexity depends on numeric value of input (not number of inputs) is called Pseudo-polynomial algorithm.  
For example, consider the problem of counting frequencies of all elements in an array of positive numbers. A pseudo-polynomial time solution for this is to first find the maximum value, then iterate from 1 to maximum value and for each value, find its frequency in array. This solution requires time according to maximum value in input array, therefore pseudo-polynomial. On the other hand, an algorithm whose time complexity is only based on number of elements in array (not value) is considered as polynomial time algorithm.

# 56. K’th Smallest/Largest Element in Unsorted Array

**kthSmallest(arr[0..n-1], k)**

**1)** Divide arr[] into ⌈n/5⌉ groups where size of each group is 5

except possibly the last group which may have less than 5 elements.

**2)** Sort the above created ⌈n/5⌉ groups and find median

of all groups. Create an auxiliary array 'median[]' and store medians

of all ⌈n/5⌉ groups in this median array.

// Recursively call this method to find median of median[0..⌈n/5⌉-1]

**3)** medOfMed = kthSmallest(median[0..⌈n/5⌉-1], ⌈n/10⌉)

**4)** Partition arr[] around medOfMed and obtain its position.

pos = partition(arr, n, medOfMed)

**5)** If pos == k return medOfMed

**6)** If pos > k return kthSmallest(arr[l..pos-1], k)

**7)** If pos < k return kthSmallest(arr[pos+1..r], k-pos+l-1)

57. Given an array of n elements, where each element is at most k away from its target position, devise an algorithm that sorts in O(n log k) time. For example, let us consider k is 2, an element at index 7 in the sorted array, can be at indexes 5, 6, 7, 8, 9 in the given array.

**Examples:**

Input : arr[] = {6, 5, 3, 2, 8, 10, 9}

k = 3

Output : arr[] = {2, 3, 5, 6, 8, 9, 10}

Input : arr[] = {10, 9, 8, 7, 4, 70, 60, 50}

k = 4

Output : arr[] = {4, 7, 8, 9, 10, 50, 60, 70}

Ans: use insertion sort

58. Which sorting algorithm makes least number of memory writes?

Ans: Cycle sort and selection sort. Because they make O(n) swaps only. While other fast algorithms like quick sort performs too many swaps in worst case.

59. Find the intersection of two sorted arrays.

Eg Arr1 = [1,2,3,6,7,7,8]

Arr2 = [3,4,7,8]

O/P = 3,7,8

Hint:

Use hashing to count each array’s frequency and then the min(freq1[i],freq2[i]) are the intersections for each 0<=i<=max(arr1 || arr2)

**Questions on C/C++**

1.

# Write a URL in a C++ program, does it affect the code?

**ANS:**

int main()

{

/\*\* some code\*/

//a url like::

<https://www.google.co.in/>

/\*again some code\*/

/\*

the http: will be treated as a label while rest of the url will be treated as a comment.

\*/

}

1. Represent a decimal number into binary format without using division operator.

Ans: take & with 1 to extract the last bit.

1. Given a tridiagonal matrix, store it in a single dimension array and again represent it in matrix form.

Ans : number of elements = 3n-2 for nxn matrix. 2\*i+j will be the index of each element in the array considering the row major order.

1. Calculate any nth order polynomial using atmost n multiplications and 0 divisions.

Ans: let a polynomial be ax\*\*3 + bx\*\*2 + cx + d

write it like :: ((a\*x+b)\*x+c)\*x+d

5. reverse a string. Take input from user and print reverse of the string without using any array or string datatype. Although a single integer can be used.

Ans.. Use recursion

void print()

{

char/int ch;

cin>>ch;

if ch=='\n' : return;

print();

cout<<ch;

}

**Machine Learning Questions::**  
Q1- What’s the trade-off between bias and variance?

More reading: [Bias-Variance Tradeoff (Wikipedia)](https://en.wikipedia.org/wiki/Bias-variance_tradeoff)

Bias is error due to erroneous or overly simplistic assumptions in the learning algorithm you’re using. This can lead to the model underfitting your data, making it hard for it to have high predictive accuracy and for you to generalize your knowledge from the training set to the test set.

Variance is error due to too much complexity in the learning algorithm you’re using. This leads to the algorithm being highly sensitive to high degrees of variation in your training data, which can lead your model to overfit the data. You’ll be carrying too much noise from your training data for your model to be very useful for your test data.

The bias-variance decomposition essentially decomposes the learning error from any algorithm by adding the bias, the variance and a bit of irreducible error due to noise in the underlying dataset. Essentially, if you make the model more complex and add more variables, you’ll lose bias but gain some variance — in order to get the optimally reduced amount of error, you’ll have to tradeoff bias and variance. You don’t want either high bias or high variance in your model.

Q2- What is the difference between supervised and unsupervised machine learning?

More reading: [What is the difference between supervised and unsupervised machine learning? (Quora)](https://www.quora.com/What-is-the-difference-between-supervised-and-unsupervised-learning-algorithms)

Supervised learning requires training labeled data. For example, in order to do classification (a supervised learning task), you’ll need to first label the data you’ll use to train the model to classify data into your labeled groups. Unsupervised learning, in contrast, does not require labeling data explicitly.

Q3- How is KNN different from k-means clustering?

More reading: [How is the k-nearest neighbor algorithm different from k-means clustering? (Quora)](https://www.quora.com/How-is-the-k-nearest-neighbor-algorithm-different-from-k-means-clustering)

K-Nearest Neighbors is a supervised classification algorithm, while k-means clustering is an unsupervised clustering algorithm. While the mechanisms may seem similar at first, what this really means is that in order for K-Nearest Neighbors to work, you need labeled data you want to classify an unlabeled point into (thus the nearest neighbor part). K-means clustering requires only a set of unlabeled points and a threshold: the algorithm will take unlabeled points and gradually learn how to cluster them into groups by computing the mean of the distance between different points.

The critical difference here is that KNN needs labeled points and is thus supervised learning, while k-means doesn’t — and is thus unsupervised learning.

Q4- Explain how a ROC curve works.

More reading: [Receiver operating characteristic (Wikipedia)](https://en.wikipedia.org/wiki/Receiver_operating_characteristic)

The ROC curve is a graphical representation of the contrast between true positive rates and the false positive rate at various thresholds. It’s often used as a proxy for the trade-off between the sensitivity of the model (true positives) vs the fall-out or the probability it will trigger a false alarm (false positives).

Q5- Define precision and recall.

More reading: [Precision and recall (Wikipedia)](https://en.wikipedia.org/wiki/Precision_and_recall)

Recall is also known as the true positive rate: the amount of positives your model claims compared to the actual number of positives there are throughout the data. Precision is also known as the positive predictive value, and it is a measure of the amount of accurate positives your model claims compared to the number of positives it actually claims. It can be easier to think of recall and precision in the context of a case where you’ve predicted that there were 10 apples and 5 oranges in a case of 10 apples. You’d have perfect recall (there are actually 10 apples, and you predicted there would be 10) but 66.7% precision because out of the 15 events you predicted, only 10 (the apples) are correct.

Q6- What is Bayes’ Theorem? How is it useful in a machine learning context?

More reading: [An Intuitive (and Short) Explanation of Bayes’ Theorem (BetterExplained)](https://betterexplained.com/articles/an-intuitive-and-short-explanation-of-bayes-theorem/)

Bayes’ Theorem gives you the posterior probability of an event given what is known as prior knowledge.

Mathematically, it’s expressed as the true positive rate of a condition sample divided by the sum of the false positive rate of the population and the true positive rate of a condition. Say you had a 60% chance of actually having the flu after a flu test, but out of people who had the flu, the test will be false 50% of the time, and the overall population only has a 5% chance of having the flu. Would you actually have a 60% chance of having the flu after having a positive test?

Bayes’ Theorem says no. It says that you have a (.6 \* 0.05) (True Positive Rate of a Condition Sample) / (.6\*0.05)(True Positive Rate of a Condition Sample) + (.5\*0.95) (False Positive Rate of a Population)  = 0.0594 or 5.94% chance of getting a flu.

Bayes’ Theorem is the basis behind a branch of machine learning that most notably includes the Naive Bayes classifier. That’s something important to consider when you’re faced with machine learning interview questions.

Q7- Why is “Naive” Bayes naive?

More reading: [Why is “naive Bayes” naive? (Quora)](https://www.quora.com/Why-is-naive-Bayes-naive?share=1)

Despite its practical applications, especially in text mining, Naive Bayes is considered “Naive” because it makes an assumption that is virtually impossible to see in real-life data: the conditional probability is calculated as the pure product of the individual probabilities of components. This implies the absolute independence of features — a condition probably never met in real life.

As a Quora commenter put it whimsically, a Naive Bayes classifier that figured out that you liked pickles and ice cream would probably naively recommend you a pickle ice cream.

Q8- Explain the difference between L1 and L2 regularization.

More reading: [What is the difference between L1 and L2 regularization? (Quora)](https://www.quora.com/What-is-the-difference-between-L1-and-L2-regularization)

L2 regularization tends to spread error among all the terms, while L1 is more binary/sparse, with many variables either being assigned a 1 or 0 in weighting. L1 corresponds to setting a Laplacean prior on the terms, while L2 corresponds to a Gaussian prior.

Q9- What’s your favorite algorithm, and can you explain it to me in less than a minute?

This type of question tests your understanding of how to communicate complex and technical nuances with poise and the ability to summarize quickly and efficiently. Make sure you have a choice and make sure you can explain different algorithms so simply and effectively that a five-year-old could grasp the basics!

Q10- What’s the difference between Type I and Type II error?

More reading: [Type I and type II errors (Wikipedia)](https://en.wikipedia.org/wiki/Type_I_and_type_II_errors)

Don’t think that this is a trick question! Many machine learning interview questions will be an attempt to lob basic questions at you just to make sure you’re on top of your game and you’ve prepared all of your bases.

Type I error is a false positive, while Type II error is a false negative. Briefly stated, Type I error means claiming something has happened when it hasn’t, while Type II error means that you claim nothing is happening when in fact something is.

A clever way to think about this is to think of Type I error as telling a man he is pregnant, while Type II error means you tell a pregnant woman she isn’t carrying a baby.

Q11- What’s a Fourier transform?

More reading: [Fourier transform (Wikipedia)](https://en.wikipedia.org/wiki/Fourier_transform)

A Fourier transform is a generic method to decompose generic functions into a superposition of symmetric functions. Or as this [more intuitive tutorial](https://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/) puts it, given a smoothie, it’s how we find the recipe. The Fourier transform finds the set of cycle speeds, amplitudes and phases to match any time signal. A Fourier transform converts a signal from time to frequency domain — it’s a very common way to extract features from audio signals or other time series such as sensor data.

Q12- What’s the difference between probability and likelihood?

More reading: [What is the difference between “likelihood” and “probability”? (Cross Validated)](https://stats.stackexchange.com/questions/2641/what-is-the-difference-between-likelihood-and-probability#2647)

Q13- What is deep learning, and how does it contrast with other machine learning algorithms?

More reading: [Deep learning (Wikipedia)](https://en.wikipedia.org/wiki/Deep_learning)

Deep learning is a subset of machine learning that is concerned with neural networks: how to use backpropagation and certain principles from neuroscience to more accurately model large sets of unlabelled or semi-structured data. In that sense, deep learning represents an unsupervised learning algorithm that learns representations of data through the use of neural nets.

Q14- What’s the difference between a generative and discriminative model?

More reading: [What is the difference between a Generative and Discriminative Algorithm? (Stack Overflow)](https://stackoverflow.com/questions/879432/what-is-the-difference-between-a-generative-and-discriminative-algorithm)

A generative model will learn categories of data while a discriminative model will simply learn the distinction between different categories of data. Discriminative models will generally outperform generative models on classification tasks.

Q15- What cross-validation technique would you use on a time series dataset?

More reading: [Using k-fold cross-validation for time-series model selection (CrossValidated)](https://stats.stackexchange.com/questions/14099/using-k-fold-cross-validation-for-time-series-model-selection)

Instead of using standard k-folds cross-validation, you have to pay attention to the fact that a time series is not randomly distributed data — it is inherently ordered by chronological order. If a pattern emerges in later time periods for example, your model may still pick up on it even if that effect doesn’t hold in earlier years!

You’ll want to do something like forward chaining where you’ll be able to model on past data then look at forward-facing data.

* fold 1 : training [1], test [2]
* fold 2 : training [1 2], test [3]
* fold 3 : training [1 2 3], test [4]
* fold 4 : training [1 2 3 4], test [5]
* fold 5 : training [1 2 3 4 5], test [6]

Q16- How is a decision tree pruned?

More reading: [Pruning (decision trees)](https://en.wikipedia.org/wiki/Pruning_%28decision_trees%29)

Pruning is what happens in decision trees when branches that have weak predictive power are removed in order to reduce the complexity of the model and increase the predictive accuracy of a decision tree model. Pruning can happen bottom-up and top-down, with approaches such as reduced error pruning and cost complexity pruning.

Reduced error pruning is perhaps the simplest version: replace each node. If it doesn’t decrease predictive accuracy, keep it pruned. While simple, this heuristic actually comes pretty close to an approach that would optimize for maximum accuracy.

Q17- Which is more important to you– model accuracy, or model performance?

More reading: [Accuracy paradox (Wikipedia)](https://en.wikipedia.org/wiki/Accuracy_paradox)

This question tests your grasp of the nuances of machine learning model performance! Machine learning interview questions often look towards the details. There are models with higher accuracy that can perform worse in predictive power — how does that make sense?

Well, it has everything to do with how model accuracy is only a subset of model performance, and at that, a sometimes misleading one. For example, if you wanted to detect fraud in a massive dataset with a sample of millions, a more accurate model would most likely predict no fraud at all if only a vast minority of cases were fraud. However, this would be useless for a predictive model — a model designed to find fraud that asserted there was no fraud at all! Questions like this help you demonstrate that you understand model accuracy isn’t the be-all and end-all of model performance.

Q18- What’s the F1 score? How would you use it?

More reading: [F1 score (Wikipedia)](https://en.wikipedia.org/wiki/F1_score)

The F1 score is a measure of a model’s performance. It is a weighted average of the precision and recall of a model, with results tending to 1 being the best, and those tending to 0 being the worst. You would use it in classification tests where true negatives don’t matter much.

Q19- How would you handle an imbalanced dataset?

More reading: [8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset (Machine Learning Mastery)](http://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/)

An imbalanced dataset is when you have, for example, a classification test and 90% of the data is in one class. That leads to problems: an accuracy of 90% can be skewed if you have no predictive power on the other category of data! Here are a few tactics to get over the hump:

1- Collect more data to even the imbalances in the dataset.

2- Resample the dataset to correct for imbalances.

3- Try a different algorithm altogether on your dataset.

What’s important here is that you have a keen sense for what damage an unbalanced dataset can cause, and how to balance that.

Q20- When should you use classification over regression?

More reading: [Regression vs Classification (Math StackExchange)](https://math.stackexchange.com/questions/141381/regression-vs-classification)

Classification produces discrete values and dataset to strict categories, while regression gives you continuous results that allow you to better distinguish differences between individual points. You would use classification over regression if you wanted your results to reflect the belongingness of data points in your dataset to certain explicit categories (ex: If you wanted to know whether a name was male or female rather than just how correlated they were with male and female names.)

Q21- Name an example where ensemble techniques might be useful.

More reading: [Ensemble learning (Wikipedia)](https://en.wikipedia.org/wiki/Ensemble_learning)

Ensemble techniques use a combination of learning algorithms to optimize better predictive performance. They typically reduce overfitting in models and make the model more robust (unlikely to be influenced by small changes in the training data).

You could list some examples of ensemble methods, from bagging to boosting to a “bucket of models” method and demonstrate how they could increase predictive power.

Q22- How do you ensure you’re not overfitting with a model?

More reading: [How can I avoid overfitting? (Quora)](https://www.quora.com/How-can-I-avoid-overfitting)

This is a simple restatement of a fundamental problem in machine learning: the possibility of overfitting training data and carrying the noise of that data through to the test set, thereby providing inaccurate generalizations.

There are three main methods to avoid overfitting:

1- Keep the model simpler: reduce variance by taking into account fewer variables and parameters, thereby removing some of the noise in the training data.

2- Use cross-validation techniques such as k-folds cross-validation.

3- Use regularization techniques such as LASSO that penalize certain model parameters if they’re likely to cause overfitting.

Q23- What evaluation approaches would you work to gauge the effectiveness of a machine learning model?

More reading: [How to Evaluate Machine Learning Algorithms (Machine Learning Mastery)](http://machinelearningmastery.com/how-to-evaluate-machine-learning-algorithms/)

You would first split the dataset into training and test sets, or perhaps use cross-validation techniques to further segment the dataset into composite sets of training and test sets within the data. You should then implement a choice selection of performance metrics: here is a fairly [comprehensive list](http://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/). You could use measures such as the F1 score, the accuracy, and the confusion matrix. What’s important here is to demonstrate that you understand the nuances of how a model is measured and how to choose the right performance measures for the right situations.

Q24- How would you evaluate a logistic regression model?

More reading: [Evaluating a logistic regression (CrossValidated)](https://stats.stackexchange.com/questions/71517/evaluating-a-logistic-regression#71522), [Logistic Regression in Plain English](https://www.springboard.com/blog/logistic-regression-explained/)

A subsection of the question above. You have to demonstrate an understanding of what the typical goals of a logistic regression are (classification, prediction, etc.) and bring up a few examples and use cases.

Q25- What’s the “kernel trick” and how is it useful?

More reading: [Kernel method (Wikipedia)](https://en.wikipedia.org/wiki/Kernel_method)

The Kernel trick involves kernel functions that can enable in higher-dimension spaces without explicitly calculating the coordinates of points within that dimension: instead, kernel functions compute the inner products between the images of all pairs of data in a feature space. This allows them the very useful attribute of calculating the coordinates of higher dimensions while being computationally cheaper than the explicit calculation of said coordinates. Many algorithms can be expressed in terms of inner products. Using the kernel trick enables us effectively run algorithms in a high-dimensional space with lower-dimensional data.

### Machine Learning Interview Questions: Programming

These machine learning interview questions test your knowledge of programming principles you need to implement machine learning principles in practice. Machine learning interview questions tend to be technical questions that test your logic and programming skills: this section focuses more on the latter.

Q26- How do you handle missing or corrupted data in a dataset?

More reading: [Handling missing data (O’Reilly)](https://www.oreilly.com/learning/handling-missing-data)

You could find missing/corrupted data in a dataset and either drop those rows or columns, or decide to replace them with another value.

In Pandas, there are two very useful methods: isnull() and dropna() that will help you find columns of data with missing or corrupted data and drop those values. If you want to fill the invalid values with a placeholder value (for example, 0), you could use the fillna() method.

Q27- Do you have experience with Spark or big data tools for machine learning?

More reading: [50 Top Open Source Tools for Big Data (Datamation)](http://www.datamation.com/data-center/50-top-open-source-tools-for-big-data-1.html)

You’ll want to get familiar with the meaning of big data for different companies and the different tools they’ll want. Spark is the big data tool most in demand now, able to handle immense datasets with speed. Be honest if you don’t have experience with the tools demanded, but also take a look at job descriptions and see what tools pop up: you’ll want to invest in familiarizing yourself with them.

Q28- Pick an algorithm. Write the psuedo-code for a parallel implementation.

More reading: [Writing pseudocode for parallel programming (Stack Overflow)](https://stackoverflow.com/questions/5583257/writing-pseudocode-for-parallel-programming)

This kind of question demonstrates your ability to think in parallelism and how you could handle concurrency in programming implementations dealing with big data. Take a look at pseudocode frameworks such as [Peril-L](http://www.eng.utah.edu/~cs4960-01/lecture4.pdf) and visualization tools such as [Web Sequence Diagrams](https://www.websequencediagrams.com/) to help you demonstrate your ability to write code that reflects parallelism.

Q29- What are some differences between a linked list and an array?

More reading: [Array versus linked list (Stack Overflow)](https://stackoverflow.com/questions/166884/array-versus-linked-list#167016)

An array is an ordered collection of objects. A linked list is a series of objects with pointers that direct how to process them sequentially. An array assumes that every element has the same size, unlike the linked list. A linked list can more easily grow organically: an array has to be pre-defined or re-defined for organic growth. Shuffling a linked list involves changing which points direct where — meanwhile, shuffling an array is more complex and takes more memory.

Q30- Describe a hash table.

More reading: [Hash table (Wikipedia)](https://en.wikipedia.org/wiki/Hash_table)

A hash table is a data structure that produces an associative array. A key is mapped to certain values through the use of a hash function. They are often used for tasks such as database indexing.

Q31- Which data visualization libraries do you use? What are your thoughts on the best data visualization tools?

More reading: [31 Free Data Visualization Tools (Springboard)](https://www.springboard.com/blog/31-free-data-visualization-tools/)

What’s important here is to define your views on how to properly visualize data and your personal preferences when it comes to tools. Popular tools include R’s ggplot, Python’s seaborn and matplotlib, and tools such as Plot.ly and Tableau.

Related: [20 Python Interview Questions](https://www.springboard.com/blog/python-interview-questions/)

### Machine Learning Interview Questions: Company/Industry Specific

These machine learning interview questions deal with how to implement your general machine learning knowledge to a specific company’s requirements. You’ll be asked to create case studies and extend your knowledge of the company and industry you’re applying for with your machine learning skills.

Q32- How would you implement a recommendation system for our company’s users?

More reading: [How to Implement A Recommendation System? (Stack Overflow)](https://stackoverflow.com/questions/6302184/how-to-implement-a-recommendation-system#6302223)

A lot of machine learning interview questions of this type will involve implementation of machine learning models to a company’s problems. You’ll have to research the company and its industry in-depth, especially the revenue drivers the company has, and the types of users the company takes on in the context of the industry it’s in.

Q33- How can we use your machine learning skills to generate revenue?

More reading: [Startup Metrics for Startups (500 Startups)](http://www.slideshare.net/dmc500hats/startup-metrics-for-pirates-long-version)

This is a tricky question. The ideal answer would demonstrate knowledge of what drives the business and how your skills could relate. For example, if you were interviewing for music-streaming startup Spotify, you could remark that your skills at developing a better recommendation model would increase user retention, which would then increase revenue in the long run.

The startup metrics Slideshare linked above will help you understand exactly what performance indicators are important for startups and tech companies as they think about revenue and growth.

Q34- What do you think of our current data process?

More reading: [The Data Science Process Email Course – Springboard](https://www.springboard.com/resources/data-science-process/)

This kind of question requires you to listen carefully and impart feedback in a manner that is constructive and insightful. Your interviewer is trying to gauge if you’d be a valuable member of their team and whether you grasp the nuances of why certain things are set the way they are in the company’s data process based on company- or industry-specific conditions. They’re trying to see if you can be an intellectual peer. Act accordingly.

### Machine Learning Interview Questions: General Machine Learning Interest

This series of machine learning interview questions attempts to gauge your passion and interest in machine learning. The right answers will serve as a testament for your commitment to being a lifelong learner in machine learning.

Q35- What are the last machine learning papers you’ve read?

More reading: [What are some of the best research papers/books for machine learning?](https://www.quora.com/What-are-some-of-the-best-research-papers-books-for-Machine-learning)

Keeping up with the latest scientific literature on machine learning is a must if you want to demonstrate interest in a machine learning position. This overview of [deep learning in Nature](http://www.cs.toronto.edu/~hinton/absps/NatureDeepReview.pdf) by the scions of deep learning themselves (from Hinton to Bengio to LeCun) can be a good reference paper and an overview of what’s happening in deep learning — and the kind of paper you might want to cite.

Q36- Do you have research experience in machine learning?

Related to the last point, most organizations hiring for machine learning positions will look for your formal experience in the field. Research papers, co-authored or supervised by leaders in the field, can make the difference between you being hired and not. Make sure you have a summary of your research experience and papers ready — and an explanation for your background and lack of formal research experience if you don’t.

Q37- What are your favorite use cases of machine learning models?

More reading: [What are the typical use cases for different machine learning algorithms? (Quora)](https://www.quora.com/What-are-the-typical-use-cases-for-different-machine-learning-algorithms)

The Quora thread above contains some examples, such as decision trees that categorize people into different tiers of intelligence based on IQ scores. Make sure that you have a few examples in mind and describe what resonated with you. It’s important that you demonstrate an interest in how machine learning is implemented.

Q38- How would you approach the “Netflix Prize” competition?

More reading: [Netflix Prize (Wikipedia)](https://en.wikipedia.org/wiki/Netflix_Prize)

The Netflix Prize was a famed competition where Netflix offered $1,000,000 for a better collaborative filtering algorithm. The team that won called BellKor had a 10% improvement and used an ensemble of different methods to win. Some familiarity with the case and its solution will help demonstrate you’ve paid attention to machine learning for a while.

Q39- Where do you usually source datasets?

More reading: [19 Free Public Data Sets For Your First Data Science Project (Springboard)](https://www.springboard.com/blog/free-public-data-sets-data-science-project/)

Machine learning interview questions like these try to get at the heart of your machine learning interest. Somebody who is truly passionate about machine learning will have gone off and done side projects on their own, and have a good idea of what great datasets are out there. If you’re missing any, check out [Quandl](https://www.quandl.com/) for economic and financial data, and [Kaggle’s Datasets](https://www.kaggle.com/datasets) collection for another great list.

Q40- How do you think Google is training data for self-driving cars?

More reading: [Waymo Tech](https://waymo.com/tech/)

Machine learning interview questions like this one really test your knowledge of different machine learning methods, and your inventiveness if you don’t know the answer. Google is currently using [recaptcha](https://www.google.com/recaptcha) to source labeled data on storefronts and traffic signs. They are also building on training data collected by Sebastian Thrun at GoogleX — some of which was obtained by his grad students driving buggies on desert dunes!

Q41- How would you simulate the approach AlphaGo took to beat Lee Sidol at Go?

More reading: [Mastering the game of Go with deep neural networks and tree search (Nature)](http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html)

AlphaGo beating Lee Sidol, the best human player at Go, in a best-of-five series was a truly seminal event in the history of machine learning and deep learning. The Nature paper above describes how this was accomplished with “Monte-Carlo tree search with deep neural networks that have been trained by supervised learning, from human expert games, and by reinforcement learning from games of self-play.”