# **Python Training**

# 

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Days** | **Topics** | **Sub Topics** | **Questions** | | **Video Links** |
| 01 June 2020 | Arrays and Multi Dimension array | Array creation and its various operations. | Given an array A[O ... n - 1], where each clement of the array represents a vote in the election. Assume that each vole is given as an integer representing the ID of the chosen candidate. Give an algorithm for determining who wins the election. | | Array creation <https://www.youtube.com/watch?v=6a39OjkCN5I>  Array search  <https://www.youtube.com/watch?v=9c9qhIcB3NA>  Sorting  Bubble (<https://www.youtube.com/watch?v=Vca808JTbI8>)  Merge (<https://www.youtube.com/watch?v=_trEkEX_-2Q>)  Selection (<https://www.youtube.com/watch?v=5KjapFQNxUo>) |
| Given an array A[] of n numbers and another number x, determines whether or not there exist two elements in S whose sum is exactly x. | |
| Given a matrix of m x n elements (m rows, n columns), return all elements of the matrix in spiral order.  Example 1:  Input:  [  [ 1, 2, 3 ],  [ 4, 5, 6 ],  [ 7, 8, 9 ]  ]  Output: [1,2,3,6,9,8,7,4,5]  Example 2:  Input:  [  [1, 2, 3, 4],  [5, 6, 7, 8],  [9,10,11,12]  ]  Output: [1,2,3,4,8,12,11,10,9,5,6,7] | |
| Implement next permutation, which rearranges numbers into the lexicographically next greater permutation of numbers.If such arrangement is not possible, it must rearrange it as the lowest possible order (ie, sorted in ascending order).  The replacement must be in-place and use only constant extra memory.  Here are some examples. Inputs are in the left-hand column and its corresponding outputs are in the right-hand column.  1,2,3 → 1,3,2  3,2,1 → 1,2,3  1,1,5 → 1,5,1 | |
| 02 June 2020 | Search Algorithms | Linear Search Algorithm  Binary Search Algorithm | Given an array of positive integers. All numbers occur even number of times except one number which occurs odd number of times. Find the number in O(n) time & constant space | | <https://www.youtube.com/watch?v=qBGLYzFF1aQ> |
| Given a sorted array of n integers that has been rotated an unknown number of times, give a O(logn) algorithm that finds an clement in the array. Example: Find 5 in array (1S1619 20 25 13 4 5 7 10 14) Output: 8 (the index of 5 in the array) | |
| Implement Recursive and Iterative binary search an element in a given array | |
| 03 June 2020 | Sort Algorithms | Sort Algorithms  Introduction  Selection Sort Algorithm  Bubble Sort Algorithm  Insertion Sort Algorithm | Minimum number of swaps required to sort an array of first N number using bubble sort | | <https://www.youtube.com/watch?v=Nkw6Jg_Gi4w&list=PLj8W7XIvO93rJHSYzkk7CgfiLQRUEC2Sq> |
| Implement Insertion sort algorithm | |
| 04 June 2020 | Sort Algorithms | Merge Sort Algorithm  Quick Sort Algorithm  Comparison | Implement merge sort and sort a given linked list using merge sort | | <https://www.youtube.com/watch?v=7h1s2SojIRw>  <https://www.youtube.com/watch?v=jOyYaBHyN28> |
| Implement Quick sort and Discuss the role of pivot | |
| 05 June 2020 | String and Recursion | String, Recursion and its related operations | Check whether two strings are anagram of each other | | <https://www.youtube.com/watch?v=k9TUPpGqYTo>  <https://www.youtube.com/watch?v=7bfokb9zy2Q>  <https://www.youtube.com/watch?v=vTX3IwquFkc> |
| Find strings given length containing balanced parentheses using recursion  Input n=4  Output:  (( ))  ( )( )  Input n=6  Output:  ((( )))  ( )(( ))  ( ( ))( )  (( )( ))  ( )( )( ) | |
| Given a set of characters CHARS and a input string INPUT, find the minimum window in str  which will contain all the characters in CHARS in complexity O(n). For example, INPUT = ABBACBAA and  CHARS = AAB has the minimum window BAA. | |
| Give an algorithm for printing all possible permutations  of the characters in a string. Unlike combinations, two permutations are considered distinct if they contain the same characters but in a different order. For simplicity assume that each occurrence of a repeated character is a distinct character. That is, if the input is "aaa", the output should be six repetitions of "aaa".  The permutations may be output in any order. | |
| **06 JUNE 2020 (SATURDAY) ASSESSMENT** | | | | |  |
| 08 June 2020 | Linked List | Creation of Linked List and various operation on Linked List | | Given a linked list consists of data, a next pointer and also a random pointer which points to random node of the list. Write code for cloning the list. | Linked List <https://www.youtube.com/watch?v=Bd1L64clh34>  <https://www.youtube.com/watch?v=qp8u-frRAnU> (Best) |
| Find modular node from the end: Given a singly linked list, write a function to find the first from the end whose *n%k* =0, where *n* is the number of elements in the list and *k* is an integer constant.  If *n* = 19 and *k* = 3 then we should return 161h node. |
| Reversing a linked list in pairs If you have a linked list that holds 1 -> 2 --> 3->4->X then after reverse it should return 2->1->4->3->X |
| Suppose there are two singly linked lists both of which intersect at some point and become a  single linked list. The head or start pointers of both the lists are known, but the intersecting node is not known. Also, the number of nodes in each of the lists before they intersect is unknown and may be different in each list. *list* 1 may have n nodes before it reaches the intersection point, and list 2*.* might have m nodes before it reaches the intersection point where *m* and n mny be *m* = *n,* m < n or m > n. |
| 09 June 2020  \ | Stack | Stack operation and its implementation | | Infix to Postfix Conversion, and postfix evaluation using stack. | Stack <https://www.youtube.com/watch?v=lVFnq4zbs-g&list=PL5tcWHG-UPH112e7AN7C-fwDVPVrt0wpV> (part-1)  <https://www.youtube.com/watch?v=TC7apM-xGaU&list=PL5tcWHG-UPH112e7AN7C-fwDVPVrt0wpV&index=2> (part-2)  <https://www.youtube.com/watch?v=X41iojWqQZY&list=PL5tcWHG-UPH112e7AN7C-fwDVPVrt0wpV&index=3> (Part-3) |
| Given a stack, how to reverse the elements of the stack using only stack operations (push & pop) |
| Given n non-negative integers representing the histogram's bar height where the width of each bar is 1, find the area of largest rectangle in the histogram. For example, consider the following histogram with 7 bars of heights {6, 2, 5, 4, 5, 1, 6}. The largest possible rectangle possible is 12 (see the below figure, the max area rectangle is highlighted in red) Note: Solve the problem in both complexity O(nlog) and O(n) |
| Given a non-negative integer num represented as a string, remove k digits from the number so that the new number is the smallest possible.  Note:  The length of num is less than 10002 and will be ≥ k.  The given num does not contain any leading zero.  Example 1:  Input: num = "1432219", k = 3  Output: "1219"  Explanation: Remove the three digits 4, 3, and 2 to form the new number 1219 which is the smallest.  Example 2:  Input: num = "10200", k = 1  Output: "200"  Explanation: Remove the leading 1 and the number is 200. Note that the output must not contain leading zeroes.  Example 3:  Input: num = "10", k = 2  Output: "0"  Explanation: Remove all the digits from the number and it is left with nothing which is 0. Note: Time Complexity: O(kn) , where k is the count of numbers to be removed and n is the length of string. |  |
| 10 June 2020 | Queue | Queue and its operation and implementation | | Queue Implementation using two stacks | Queue  <https://www.youtube.com/watch?v=bnm5_GH04fM> (Part-1)  <https://www.youtube.com/watch?v=wkPMom77Hqk> (Part-2) |
| Given o string, write a Python method to check whether it is a palindrome or nor using doubly  ended queue. |
| 11 June 2020 | Hashing and Hash Table | Introduction  Hashing  Hash Table | | Check whether the given linked list is either NULL-terminated or ends in a cycle (cyclic) using hash table.  Note :Time Complexity Shoud be O(n). |  |
| Rajiv and Nitish had a fight because Rajiv was annoying Nitish with his question. Rajiv being a genius in arrays gave Nitish an array of natural numbers A of length N with elements A1, A2, . . ., AN. Nitish has to find the total amount of perfect pairs in the array.    A perfect pair (Ai, Aj) is a pair where (Ai + Aj) is a perfect square or a perfect cube and i ≠ j.    Since Rajiv and Nitish are not talking with each other after the fight you have been given the question to solve  and inturn make both of them a perfect pair again.    NOTE :- A pair (Ai, Aj) and (Aj, Ai) are same and not to be counted twice.    Input  The first line on the input contains the a single integer T denoting the number of test cases. The first line of each test case  contains a single integer N. The second line contains N space-separated integers A1, A2, . . ., AN.  Output  For each test case, print a single line containing a single integer denoting the total number of perfect pairs.  Constraints  1 ≤ T ≤ 10  1 ≤ N ≤ 105  1 ≤ Ai ≤ 103  SAMPLE INPUT  2  5  1 2 3 4 5  4  1 4 5 8  SAMPLE OUTPUT  3  2  Explanation  In first testcase pairs (1, 3), (3, 5) and (4,5) give values 4, 8, 9 and form perfect pairs.  In the second testcase pairs (1, 8), (4, 5) give values 9, 9 and form perfect pairs. | <https://www.youtube.com/watch?v=KyUTuwz_b7Q>  <https://www.youtube.com/watch?v=xo9Vsfzfg8g>  <https://www.youtube.com/watch?v=wWgIAphfn2U>  <https://www.youtube.com/watch?v=Dk57JonwKNk> |
| 12 June 2020 | Tree | Introduction  Trees,  Binary Trees,  Binary Search Tree  Tree Traversal : In order, Preorder, post  order | | Count the number of leaf nodes | <https://www.youtube.com/watch?v=6oL-0TdVy28> <https://www.youtube.com/watch?v=BDw8zzy3QiY>  <https://www.youtube.com/watch?v=YlgPi75hIBc> |
| Construct a Binary Tree and print the following results:  (a) Is it Complete binary search true  (b) Is it strict binary search tree  (c ) height of the Binary tree  (d) preorder  (e ) post order  (f) inorder  (g) level order |
| Give an algorithm for finding the diameter of the binary tree. The diameter of a tree (sometimes  called the *width)* is the number of nodes on the longest path between two leaves in the tree. |
| Give an algorithm for finding the vertical sum of a binary tree. For example,  The tree has 5 vertical lines  Vertical-1: nodes-4 =>vertical sum is 4  Vertical-2: nodes-2 =>vertical sum is 2  Vertical -3: nodes- 1,5,6 => vertical sum is 1 + 5 + *6 =*12  Vertical-4: nodes-3 => vertical sum is 3  Vert.ical-5: nodes-7 =>vertical sum is 7  We need to output: 4 2 12 3 7 |
| **13 JUNE 2020 ASSESSMENT-2** | | | | |  |
| 15 June 2020 | Graph | Graph its related operations, Applications of graphs | | Detect Cycles in Directed graph | <https://www.youtube.com/watch?v=HDUzBEG1GlA> <https://www.youtube.com/watch?v=k1kLCB7AZbM>  <https://www.youtube.com/watch?v=joqmqvHC_Bo>  <https://www.youtube.com/watch?v=zaBhtODEL0w> |
| Count simple paths for a given graph *G* has simple path from sources to destination d? Assume the graph is represented using the adjacent matrix. |
| Implement BFS and DFS |
| Implement Dijkstra’s shortest path algorithm and print the shortest path and cost of path |
| 16 June 2020 | Greedy Programming and complexity analysis |  | | Program for Greedy Algorithm to find Minimum number of Coins  Given a value V, if we want to make change for V Rs, and we have infinite supply of each of the  denominations in Indian  currency, i.e., we have infinite supply of { 1, 2, 5, 10, 20, 50, 100, 500, 1000} valued coins/notes, what is the  minimum  number of coins and/or notes needed to make the change?  Examples:  Input: V = 70  Output: 2  We need a 50 Rs note and a 20 Rs note.  Input: V = 121  Output: 3  We need a 100 Rs note, a 20 Rs note and a  1 Rs coin. | <https://www.youtube.com/watch?v=QvSIAB27Vdk>  <https://www.youtube.com/watch?v=lFSOj4qce4w>  <https://www.youtube.com/watch?v=KXAbAa1mieU>  <https://www.youtube.com/watch?v=aQsJSUXwFco> (Theoretical - time and space complexity) |
| Implement fractional kanpsack problem using greedy approach  Input: items[] = [ [60, 10], [100, 20], [120, 30] ]  Knapsack Capacity(capacity) = 50  Output: Maximum possible value = 240  Explanation: By taking full items of 10 kg, 20 kg and 2/3rd of last item of 30 kg. Total value = 60 + 100 +  120\*2/3 = 240 |  |
|  |  |  | |  |
| Gives the Analysis of Time and space complexity of BFS and BFS |
| 17 June 2020 | Dynamic Programming and Complexity analysis |  | | LCS Problem Statement: Given two sequences, find the length of longest subsequence  present in both of them.  Examples:  LCS for input Sequences “ABCDGH” and “AEDFHR” is “ADH” of length 3.  LCS for input Sequences “AGGTAB” and “GXTXAYB” is “GTAB” of length 4. | <https://www.youtube.com/watch?v=HgUOWB0StNE>  <https://www.youtube.com/watch?v=IRwVmTmN6go&t=280s> |
| Given a rod of length n inches and an array of prices that contains prices of all pieces of size smaller than n.  Determine the maximum value obtainable by cutting up the rod and selling the pieces.  For example, if length of the rod is 8 and the values of different pieces are given as following,  then the maximum obtainable value is 22 (by cutting in two pieces of lengths 2 and 6)  length | 1 2 3 4 5 6 7 8  --------------------------------------------  price | 1 5 8 9 10 17 17 20  And if the prices are as following, then the maximum obtainable value is 24 (by cutting in eight pieces of  length 1)  length | 1 2 3 4 5 6 7 8  --------------------------------------------  price | 3 5 8 9 10 17 17 20 |
| Gives the Analysis of time and space complexity of Insertion sort, Merge sort and Quick Sort |  |
| 18 June 2020 | OOPS Concept | Class, Object, constructor, in build functions Relationships between objects, abstract class, Inheritance, types of inheritance | | (a) Write a program in Python for the class Person using \_\_init\_\_(self) , the class constructor.  (b) Add the method: \_\_del\_\_(self) to the above definition. (Destructor)  (c) Add a class attribute- counter and object attributes- self.name and self.age. Increase counter  by 1 in the constructor and initialize name and age in constructor | <https://www.youtube.com/watch?v=8O5kX73OkIY>  <https://www.youtube.com/watch?v=ic6wdPxcHc0>  <https://www.youtube.com/watch?v=Cn7AkDb4pIU>  <https://www.youtube.com/watch?v=SRu1GAfr3LA> |
| (a) Write a program in Python that uses class to store the name and marks of stude  [Note: use list to store marks in 3 subjects]  (b) Create two instances of this class and make use of instance method to display marks.  (c) Write code in Python to show the use of the following built-in functions:  Getattr(), setattr(), hasattr() and delattr() |
| Write an “abstract” class, Box, and use it to define some methods which any box object should have: add, for adding any number of items to the box, empty, for taking all the items out of the box and returning them as a list, and count, for counting the items which are currently in the box. Write a simple Item class which has a name attribute and a value attribute – you can assume that all the items you will use will be Item objects. Now write two subclasses of Box which use different underlying collections to store items: ListBox should use a list, and DictBox should use a dict. |
| (a) Write a program to define the same function area to compute the area of circle and rectangle  depending upon the number of arguments passed. (Method overloading)  (b) Write the Python code for method overriding:  Class Person: It has an instance method display (self)  Class Student: It also has an instance method display (self)  (c). Write Python code to show the usage of the built-in class attributes like  \_\_doc\_\_,\_\_dict\_\_etc.  3  (a) Write a program in |  |
| Write a function, repack\_boxes, which takes any number of boxes as parameters, gathers up all the items they contain, and redistributes them as evenly as possible over all the boxes. Order is unimportant. There are multiple ways of doing this. Test your code with a ListBox with 20 items, a ListBox with 9 items and a DictBox with 5 items. You should end up with two boxes with 11 items each, and one box with 12 items. |  |
| 19 June 2020 | OOPS Concepts | Polymorphism, access modifiers, class decorators | | (a) Write a program to define the same function area to compute the area of circle and rectangle  depending upon the number of arguments passed. (Method overloading)  (b) Write the Python code for method overriding:  Class Person: It has an instance method display (self)  Class Student: It also has an instance method display (self)  (c). Write Python code to show the usage of the built-in class attributes like  \_\_doc\_\_,\_\_dict\_\_etc. | <https://www.youtube.com/watch?v=8O5kX73OkIY>  <https://www.youtube.com/watch?v=ic6wdPxcHc0>  <https://www.youtube.com/watch?v=Cn7AkDb4pIU>  <https://www.youtube.com/watch?v=SRu1GAfr3LA> |
| Create a class called Numbers, which has a single class attribute called MULTIPLIER, and a constructor which takes the parameters x and y (these should all be numbers).  (a) Write a method called add which returns the sum of the attributes x and y.  (b) Write a class method called multiply, which takes a single number parameter a and returns the product of a and MULTIPLIER.  (c) Write a static method called subtract, which takes two number parameters, b and c, and returns b - c.  (d) Write a method called value which returns a tuple containing the values of x and y. Make this method into a property, and write a setter and a deleter for manipulating the values of x and y. |  |
| **20 JUNE 2020 ASSESSMENT-3** | | | | |  |