1. Program for array rotation
2. Reversal algorithm for array rotation
3. Block swap algorithm for array rotation
4. Program to cyclically rotate an array by one
5. KMP Algorithm for Pattern Searching
6. Search an element in a sorted and rotated array
7. Given a sorted and rotated array, find if there is a pair with a given sum
8. Find maximum value of Sum( i\*arr[i]) with only rotations on given array allowed
9. Maximum sum of i\*arr[i] among all rotations of a given array
10. Find the Rotation Count in Rotated Sorted array
11. Quickly find multiple left rotations of an array
12. Find the minimum element in a sorted and rotated array
13. Reversal algorithm for right rotation of an array
14. Find a rotation with maximum hamming distance
15. Queries on Left and Right Circular shift on array
16. Print left rotation of array in O(n) time and O(1) space
17. Find element at given index after a number of rotations
18. Split the array and add the first part to the end
19. Rearrange an array such that arr[i] = i
20. Write a program to reverse an array or string
21. Rearrange array such that arr[i] >= arr[j] if i is even and arr[i]<=arr[j] if i is odd and j < i
22. Rearrange positive and negative numbers in O(n) time and O(1) extra space
23. Rearrange array in alternating positive & negative items with O(1) extra space | Set 1
24. Move all zeroes to end of array
25. Move all zeroes to end of array | Set-2 (Using single traversal)
26. Minimum swaps required to bring all elements less than or equal to k together
27. Rearrange positive and negative numbers using inbuilt sort function
28. Rearrange array such that even positioned are greater than odd
29. Rearrange an array in order – smallest, largest, 2nd smallest, 2nd largest, ..
30. Double the first element and move zero to end
31. Reorder an array according to given indexes
32. Rearrange positive and negative numbers with constant extra space
33. Arrange given numbers to form the biggest number
34. Rearrange an array such that ‘arr[j]’ becomes ‘i’ if ‘arr[i]’ is ‘j’
35. Rearrange an array in maximum minimum form | Set 1
36. Rearrange an array in maximum minimum form | Set 2 (O(1) extra space)
37. Move all negative numbers to beginning and positive to end with constant extra space
38. Move all negative elements to end in order with extra space allowed
39. Rearrange array such that even index elements are smaller and odd index elements are greater
40. Positive elements at even and negative at odd positions
41. Replace every array element by multiplication of previous and next
42. Shuffle a given array
43. Segregate even and odd numbers
44. Kth Smallest/Largest Element in Unsorted Array | Set 1
45. K’th Smallest/Largest Element in Unsorted Array | Set 2 (Expected Linear Time)
46. K’th Smallest/Largest Element in Unsorted Array | Set 3 (Worst Case Linear Time)
47. K’th Smallest/Largest Element using STL
48. k largest(or smallest) elements in an array | added Min Heap method
49. Kth smallest element in a row-wise and column-wise sorted 2D array | Set 1
50. Program to find largest element in an array
51. Find the largest three elements in an array
52. Find all elements in array which have at-least two greater elements
53. Program for Mean and median of an unsorted array
54. Median of Stream of Running Integers using STL
55. Minimum product of k integers in an array of positive Integers
56. K-th Largest Sum Contiguous Subarray
57. K maximum sum combinations from two arrays
58. K maximum sums of overlapping contiguous sub-arrays
59. K maximum sums of non-overlapping contiguous sub-arrays
60. k smallest elements in same order using O(1) extra space
61. Find k pairs with smallest sums in two arrays
62. k-th smallest absolute difference of two elements in an array
63. Find Second largest element in an array
64. Find k numbers with most occurrences in the given array
65. Find the smallest and second smallest elements in an array
66. Find the smallest missing number
67. Maximum sum such that no two elements are adjacent
68. Maximum and minimum of an array using minimum number of comparisons
69. MO’s Algorithm
70. Sqrt (or Square Root) Decomposition Technique | Set 1 (Introduction)
71. Sparse Table
72. Range sum query using Sparse Table
73. Range Minimum Query (Square Root Decomposition and Sparse Table)
74. Range Queries for Frequencies of array elements
75. Constant time range add operation on an array
76. Range LCM Queries
77. GCDs of given index ranges in an array
78. Queries for GCD of all numbers of an array except elements in a given range
79. Number of elements less than or equal to a given number in a given subarray
80. Number of elements less than or equal to a given number in a given subarray | Set 2 (Including Updates)
81. Queries for counts of array elements with values in given range
82. Queries for decimal values of subarrays of a binary array
83. Count elements which divide all numbers in range L-R
84. Number whose sum of XOR with given array range is maximum
85. XOR of numbers that appeared even number of times in given Range
86. Array range queries over range queries
87. Array range queries for searching an element
88. Array range queries for elements with frequency same as value
89. Maximum Occurrence in a Given Range
90. Number of indexes with equal elements in given range
91. Merge Sort Tree for Range Order Statistics
92. Total numbers with no repeated digits in a range
93. Difference Array | Range update query in O(1)
94. Largest Sum Contiguous Subarray
95. Maximum profit by buying and selling a share at most twice
96. Find the subarray with least average
97. Find the minimum distance between two numbers
98. Minimize the maximum difference between the heights
99. Minimum number of jumps to reach end
100. Dynamic Programming | Set 14 (Maximum Sum Increasing Subsequence)
101. Smallest subarray with sum greater than a given value
102. Find maximum average subarray of k length
103. Count minimum steps to get the given desired array
104. Number of subsets with product less than k
105. Find minimum number of merge operations to make an array palindrome
106. Find the smallest positive integer value that cannot be represented as sum of any subset of a given array
107. Size of The Subarray With Maximum Sum
108. Find minimum difference between any two elements
109. Space optimization using bit manipulations
110. Longest Span with same Sum in two Binary arrays
111. Alternative Sorting
112. Sort a nearly sorted (or K sorted) array
113. Sort an array according to absolute difference with given value
114. Sort an array in wave form
115. Merge an array of size n into another array of size m+n
116. Sort an array which contain 1 to n values
117. Sort 1 to N by swapping adjacent elements
118. Sort an array containing two types of elements
119. Sort elements by frequency | Set 1
120. Count Inversions in an array | Set 1 (Using Merge Sort)
121. Two elements whose sum is closest to zero
122. Shortest Un-ordered Subarray
123. Minimum number of swaps required to sort an array
124. Union and Intersection of two sorted arrays
125. Find Union and Intersection of two unsorted arrays
126. Sort an array of 0s, 1s and 2s
127. Find the Minimum length Unsorted Subarray, sorting which makes the complete array sorted
128. Median in a stream of integers (running integers)
129. Count the number of possible triangles
130. Find number of pairs (x, y) in an array such that x^y > y^x
131. Count all distinct pairs with difference equal to k
132. Print All Distinct Elements of a given integer array
133. Construct an array from its pair-sum array
134. Merge two sorted arrays with O(1) extra space
135. Product of maximum in first array and minimum in second
136. Search, insert and delete in an unsorted array
137. Search, insert and delete in a sorted array
138. Given an array A[] and a number x, check for pair in A[] with sum as x
139. Searching in an array where adjacent differ by at most k
140. Find common elements in three sorted arrays
141. Find position of an element in a sorted array of infinite numbers
142. Find the only repetitive element between 1 to n-1
143. Find the element that appears once
144. Maximum Subarray Sum Excluding Certain Elements
145. Maximum equlibrium sum in an array
146. Equilibrium index of an array
147. Leaders in an array
148. Ceiling in a sorted array
149. Majority Element
150. Check for Majority Element in a sorted array
151. Check if an array has a majority element
152. Two Pointers Technique
153. Find a peak element
154. Find the two repeating elements in a given array
155. Find a Fixed Point in a given array
156. Find sub-array with given sum
157. Maximum triplet sum in array
158. Smallest Difference Triplet from Three arrays
159. Find a triplet that sum to a given value
160. Find all triplets with zero sum
161. Rotate Matrix Elements
162. Inplace rotate square matrix by 90 degrees | Set 1
163. Rotate a matrix by 90 degree without using any extra space | Set 2
164. Rotate a Matrix by 180 degree
165. Turn an image by 90 degree
166. Rotate each ring of matrix anticlockwise by K elements
167. Check if all rows of a matrix are circular rotations of each other
168. Sort the given matrix
169. Find the row with maximum number of 1s
170. Find median in row wise sorted matrix
171. Matrix Multiplication | Recursive
172. Program to multiply two matrices
173. Program for scalar multiplication of a matrix
174. Program to print Lower triangular and Upper triangular matrix of an array
175. Find distinct elements common to all rows of a matrix
176. Print a given matrix in spiral form
177. Find maximum element of each row in a matrix
178. Find unique elements in a matrix
179. Shift matrix elements row-wise by k
180. Different Operations on Matrices
181. Print a given matrix in counter-clock wise spiral form
182. Swap major and minor diagonals of a square matrix
183. Maximum path sum in matrix
184. Squares of Matrix Diagonal Elements
185. Move matrix elements in given direction and add elements with same value
186. Subarray/Substring vs Subsequence and Programs to Generate them
187. A Product Array Puzzle
188. Number of subarrays with given product
189. Linked List vs Array
190. Check if array elements are consecutive | Added Method 3
191. Find whether an array is subset of another array | Added Method 3
192. Implement two stacks in an array
193. Find relative complement of two sorted arrays
194. Minimum increment by k operations to make all elements equal
195. Minimize (max(A[i], B[j], C[k]) – min(A[i], B[j], C[k])) of three different sorted arrays
196. Introduction to Linked List
197. Linked List vs Array
198. Linked List Insertion
199. Linked List Deletion (Deleting a given key)
200. Linked List Deletion (Deleting a key at given position)
201. Write a function to delete a Linked List
202. Find Length of a Linked List (Iterative and Recursive)
203. Search an element in a Linked List (Iterative and Recursive)
204. Write a function to get Nth node in a Linked List
205. Nth node from the end of a Linked List
206. Print the middle of a given linked list
207. Write a function that counts the number of times a given int occurs in a Linked List
208. Detect loop in a linked list
209. Find length of loop in linked list
210. Function to check if a singly linked list is palindrome
211. Remove duplicates from a sorted linked list
212. Remove duplicates from an unsorted linked list
213. Swap nodes in a linked list without swapping data
214. Pairwise swap elements of a given linked list
215. Move last element to front of a given Linked List
216. Intersection of two Sorted Linked Lists
217. Intersection point of two Linked Lists.
218. QuickSort on Singly Linked List
219. Segregate even and odd nodes in a Linked List
220. Reverse a linked list
221. Circular Linked List Introduction and Applications,
222. Circular Linked List Traversal
223. Split a Circular Linked List into two halves
224. Sorted insert for circular linked list
225. Check if a linked list is Circular Linked List
226. Convert a Binary Tree to a Circular Doubly Link List
227. Circular Singly Linked List | Insertion
228. Deletion from a Circular Linked List
229. Circular Queue | Set 2 (Circular Linked List Implementation)
230. Count nodes in Circular linked list
231. Josephus Circle using circular linked list
232. Convert singly linked list into circular linked list
233. Circular Linked List | Set 1 (Introduction and Applications)
234. Circular Linked List | Set 2 (Traversal)
235. Implementation of Deque using circular array
236. Exchange first and last nodes in Circular Linked List
237. Doubly Linked List Introduction and Insertion
238. Delete a node in a Doubly Linked List
239. Reverse a Doubly Linked List
240. The Great Tree-List Recursion Problem.
241. Copy a linked list with next and arbit pointer
242. QuickSort on Doubly Linked List
243. Swap Kth node from beginning with Kth node from end in a Linked List
244. Merge Sort for Doubly Linked List
245. Create a Doubly Linked List from a Ternary Tree
246. Find pairs with given sum in doubly linked list
247. Insert value in sorted way in a sorted doubly linked list
248. Delete a Doubly Linked List node at a given position
249. Count triplets in a sorted doubly linked list whose sum is equal to a given value x
250. Remove duplicates from a sorted doubly linked list
251. Delete all occurrences of a given key in a doubly linked list
252. Remove duplicates from an unsorted doubly linked list
253. Sort the biotonic doubly linked list
254. Sort a k sorted doubly linked list
255. Convert a given Binary Tree to Doubly Linked List | Set
256. Program to find size of Doubly Linked List
257. Sorted insert in a doubly linked list with head and tail pointers
258. Large number arithmetic using doubly linked list
259. Rotate Doubly linked list by N nodes
260. Priority Queue using doubly linked list
261. Reverse a doubly linked list in groups of given size
262. Doubly Circular Linked List | Set 1 (Introduction and Insertion)
263. Doubly Circular Linked List | Set 2 (Deletion)
264. Reverse a stack without using extra space in O(n)
265. An interesting method to print reverse of a linked list
266. Linked List representation of Disjoint Set Data Structures
267. Sublist Search (Search a linked list in another list)
268. How to insert elements in C++ STL List ?
269. Unrolled Linked List | Set 1 (Introduction)
270. A Programmer’s approach of looking at Array vs. Linked List
271. How to write C functions that modify head pointer of a Linked List?
272. Given a linked list which is sorted, how will you insert in sorted way
273. Can we reverse a linked list in less than O(n)?
274. Practice questions for Linked List and Recursion
275. Construct a Maximum Sum Linked List out of two Sorted Linked Lists having some Common nodes
276. Given only a pointer to a node to be deleted in a singly linked list, how do you delete it?
277. Why Quick Sort preferred for Arrays and Merge Sort for Linked Lists?
278. Squareroot(n)-th node in a Linked List
279. Find the fractional (or n/k – th) node in linked list
280. Find modular node in a linked list
281. Construct a linked list from 2D matrix
282. Find smallest and largest elements in singly linked list
283. Arrange consonants and vowels nodes in a linked list
284. Partitioning a linked list around a given value and If we don’t care about making the elements of the list “stable”
285. Modify contents of Linked List
286. Implement Queue using Stacks
287. Design and Implement Special Stack Data Structure | Added Space Optimized Version
288. Implement two stacks in an array
289. Implement Stack using Queues
290. Design a stack with operations on middle element
291. How to efficiently implement k stacks in a single array?
292. How to create mergable stack?
293. Design a stack that supports getMin() in O(1) time and O(1) extra space
294. Implement a stack using single queue
295. How to implement stack using priority queue or heap?
296. Create a customized data structure which evaluates functions in O(1)
297. Implement Stack and Queue using Deque
298. Standard Problems based on Stack :
299. Infix to Postfix Conversion using Stack
300. Prefix to Infix Conversion
301. Prefix to Postfix Conversion
302. Postfix to Prefix Conversion
303. Postfix to Infix
304. Convert Infix To Prefix Notation
305. The Stock Span Problem
306. Check for balanced parentheses in an expression
307. Next Greater Element
308. Next Greater Frequency Element
309. Number of NGEs to the right
310. Maximum product of indexes of next greater on left and right
311. The Celebrity Problem
312. Expression Evaluation
313. Arithmetic Expression Evalution
314. Evaluation of Postfix Expression
315. Iterative Tower of Hanoi
316. Print next greater number of Q queries
317. Operations on Stack :
318. Reverse a stack using recursion
319. Sort a stack using recursion
320. Sort a stack using a temporary stack
321. Reverse a stack without using extra space in O(n)
322. Delete middle element of a stack
323. Sorting array using Stacks
324. Delete array elements which are smaller than next or become smaller
325. Check if a queue can be sorted into another queue using a stack
326. Reverse individual words
327. Count subarrays where second highest lie before highest
328. Check if an array is stack sortable
329. Misc :
330. Iterative Postorder Traversal | Set 1 (Using Two Stacks)
331. Iterative Postorder Traversal | Set 2 (Using One Stack)
332. Merge Overlapping Intervals
333. Largest Rectangular Area in a Histogram | Set 2
334. Print ancestors of a given binary tree node without recursion
335. Reverse a string using stack
336. Program for Tower of Hanoi
337. Find maximum depth of nested parenthesis in a string
338. Find maximum of minimum for every window size in a given array
339. Length of the longest valid substring
340. Iterative Depth First Traversal of Graph
341. Minimum number of bracket reversals needed to make an expression balanced
342. Expression contains redundant bracket or not
343. Identify and mark unmatched parenthesis in an expression
344. Check if two expressions with brackets are same
345. Find index of closing bracket for a given opening bracket in an expression
346. Check for balanced parentheses in an expression
347. Balanced expression with replacement
348. Check if a given array can represent Preorder Traversal of Binary Search Tree
349. Form minimum number from given sequence
350. Find if an expression has duplicate parenthesis or not
351. Find maximum difference between nearest left and right smaller elements
352. Find next Smaller of next Greater in an array
353. Find maximum sum possible equal sum of three stacks
354. Count natural numbers whose all permutation are greater than that number
355. Delete consecutive same words in a sequence
356. Decode a string recursively encoded as count followed by substring
357. Bubble sort using two Stacks
358. Pattern Occurrences : Stack Implementation Java
359. Iterative method to find ancestors of a given binary tree
360. Stack Permutations (Check if an array is stack permutation of other)
361. Tracking current Maximum Element in a Stack
362. Check mirror in n-ary tree
363. Reverse a number using stack
364. Reversing the first K elements of a Queue
365. Reversing a Queue
366. Check if stack elements are pairwise consecutive
367. Spaghetti Stack
368. Interleave the first half of the queue with second half
369. Remove brackets from an algebraic string containing + and – operators
370. Growable array based stack
371. Range Queries for Longest Correct Bracket Subsequence
372. Handshaking Lemma and Interesting Tree Properties
373. Enumeration of Binary Trees
374. Insertion in a Binary Tree
375. Deletion in a Binary Tree
376. BFS vs DFS for Binary Tree
377. Binary Tree (Array implementation)
378. AVL with duplicate keys
379. Applications of tree data structure
380. Applications of Minimum Spanning Tree Problem
381. Continuous Tree
382. Foldable Binary Trees
383. Expression Tree
384. Evaluation of Expression Tree
385. Symmetric Tree (Mirror Image of itself)
386. Traversals :
387. Tree Traversals
388. Inorder Tree Traversal without Recursion
389. Inorder Tree Traversal without recursion and without stack!
390. Print Postorder traversal from given Inorder and Preorder traversals
391. Find postorder traversal of BST from preorder traversal
392. Find all possible binary trees with given Inorder Traversal
393. Replace each node in binary tree with the sum of its inorder predecessor and successor
394. Populate Inorder Successor for all nodes
395. Inorder Successor of a node in Binary Tree
396. Find n-th node of inorder traversal
397. Find n-th node in Postorder traversal of a Binary Tree
398. Level Order Tree Traversal
399. Level order traversal in spiral form
400. Level order traversal line by line
401. Level order traversal with direction change after every two levels
402. Reverse Level Order Traversal
403. Reverse tree path
404. Perfect Binary Tree Specific Level Order Traversal
405. Perfect Binary Tree Specific Level Order Traversal | Set 2
406. Reverse alternate levels of a perfect binary tree
407. Morris traversal for Preorder
408. Iterative Preorder Traversal
409. Iterative Postorder Traversal | Set 1 (Using Two Stacks)
410. Iterative Postorder Traversal | Set 2 (Using One Stack)
411. Postorder traversal of Binary Tree without recursion and without stack
412. Diagonal Traversal of Binary Tree
413. Iterative diagonal traversal of binary tree
414. Boundary Traversal of binary tree
415. Density of Binary Tree in One Traversal
416. Calculate depth of a full Binary tree from Preorder
417. Number of Binary Trees for given Preorder Sequence length
418. Modify a binary tree to get Preorder traversal using right pointers only
419. More >>
420. Construction & Conversion :
421. Construct Tree from given Inorder and Preorder traversals
422. Construct a tree from Inorder and Level order traversals
423. Construct Complete Binary Tree from its Linked List Representation
424. Construct a complete binary tree from given array in level order fashion
425. Construct Full Binary Tree from given preorder and postorder traversals
426. Construct Full Binary Tree using its Preorder traversal and Preorder traversal of its mirror tree
427. Construct a special tree from given preorder traversal
428. Construct tree from ancestor matrix
429. Construct Ancestor Matrix from a Given Binary Tree
430. Construct Special Binary Tree from given Inorder traversal
431. Construct Binary Tree from given Parent Array representation
432. Construct a Binary Tree from Postorder and Inorder
433. Create a Doubly Linked List from a Ternary Tree
434. Creating a tree with Left-Child Right-Sibling Representation
435. Prufer Code to Tree Creation
436. If you are given two traversal sequences, can you construct the binary tree?
437. Construct the full k-ary tree from its preorder traversal
438. Construct Binary Tree from String with bracket representation
439. Linked complete binary tree & its creation
440. Convert a given Binary Tree to Doubly Linked List | Set 1
441. Convert a given Binary Tree to Doubly Linked List | Set 2
442. Convert a given Binary Tree to Doubly Linked List | Set 3
443. Convert a given Binary Tree to Doubly Linked List | Set 4
444. Convert an arbitrary Binary Tree to a tree that holds Children Sum Property
445. Convert left-right representation of a binary tree to down-right
446. Convert a given tree to its Sum Tree
447. Change a Binary Tree so that every node stores sum of all nodes in left subtree
448. Write an Efficient Function to Convert a Binary Tree into its Mirror Tree
449. Convert a Binary Tree into Doubly Linked List in spiral fashion
450. Convert a Binary Tree to a Circular Doubly Link List
451. Convert a tree to forest of even nodes
452. Convert a given Binary tree to a tree that holds Logical AND property
453. Convert Ternary Expression to a Binary Tree
454. Flip Binary Tree
455. Minimum swap required to convert binary tree to binary search tree
456. Checking & Printing :
457. Check for Children Sum Property in a Binary Tree
458. Check if a given Binary Tree is SumTree
459. Check sum of Covered and Uncovered nodes of Binary Tree
460. Check if two nodes are cousins in a Binary Tree
461. Check if all leaves are at same level
462. Check if removing an edge can divide a Binary Tree in two halves
463. Check if given Preorder, Inorder and Postorder traversals are of same tree
464. Given level order traversal of a Binary Tree, check if the Tree is a Min-Heap
465. Check if leaf traversal of two Binary Trees is same?
466. Check if a given Binary Tree is SumTree
467. Check whether a given binary tree is perfect or not
468. Check whether a binary tree is a full binary tree or not
469. Check whether a binary tree is a full binary tree or not | Iterative Approach
470. Check whether a given Binary Tree is Complete or not | Set 1 (Iterative Solution)
471. Check if a given Binary Tree is height balanced like a Red-Black Tree
472. Check if a binary tree is subtree of another binary tree | Set 2
473. Check if a Binary Tree (not BST) has duplicate values
474. Check if a Binary Tree contains duplicate subtrees of size 2 or more
475. Check if a given graph is tree or not
476. Check if two trees are Mirror
477. Iterative method to check if two trees are mirror of each other
478. Write Code to Determine if Two Trees are Identical
479. Iterative function to check if two trees are identical
480. Check for Symmetric Binary Tree (Iterative Approach)
481. Check if there is a root to leaf path with given sequence
482. Print middle level of perfect binary tree without finding height
483. Print cousins of a given node in Binary Tree
484. Given a binary tree, print out all of its root-to-leaf paths one per line
485. Print the longest leaf to leaf path in a Binary tree.
486. Print path from root to a given node in a binary tree
487. Print root to leaf paths without using recursion
488. Print all root to leaf paths with there relative positions
489. Print the nodes at odd levels of a tree
490. Print all full nodes in a Binary Tree
491. More >>
492. Summation :
493. Sum of all nodes in a binary tree
494. Sum of all the parent nodes having child node x
495. Find sum of all left leaves in a given Binary Tree
496. Find sum of all right leaves in a given Binary Tree
497. Find sum of all nodes of the given perfect binary tree
498. Diagonal Sum of a Binary Tree
499. Find if there is a pair in root to a leaf path with sum equals to root’s data
500. Sum of nodes on the longest path from root to leaf node
501. Remove all nodes which don’t lie in any path with sum>= k
502. Find the maximum path sum between two leaves of a binary tree
503. Find the maximum sum leaf to root path in a Binary Tree
504. Maximum sum of nodes in Binary tree such that no two are adjacent
505. Maximum sum from a tree with adjacent levels not allowed
506. Find largest subtree sum in a tree
507. Print all k-sum paths in a binary tree
508. Sum of heights of all individual nodes in a binary tree
509. Subtree with given sum in a Binary Tree
510. Count subtress that sum up to a given value x
511. Sum of nodes at maximum depth of a Binary Tree
512. Difference between sums of odd level and even level nodes of a Binary Tree
513. Find maximum level sum in Binary Tree
514. Maximum spiral sum in Binary Tree
515. Sum of nodes at k-th level in a tree represented as string
516. Sum of all leaf nodes of binary tree
517. Sum of leaf nodes at minimum level
518. Root to leaf path sum equal to a given number
519. Sum of all the numbers that are formed from root to leaf paths
520. Merge Two Binary Trees by doing Node Sum (Recursive and Iterative)
521. Vertical Sum in a given Binary Tree | Set 1
522. Vertical Sum in Binary Tree
523. Find root of the tree where children id sum for every node is given
524. Replace each node in binary tree with the sum of its inorder predecessor and successor
525. Find largest subtree sum in a tree
526. Longest Common Ancestor :
527. Lowest Common Ancestor in a Binary Tree | Set 1
528. Lowest Common Ancestor in a Binary Tree | Set 2 (Using Parent Pointer)
529. Lowest Common Ancestor in a Binary Tree | Set 3
530. Find distance between two nodes of a Binary Tree
531. Print common nodes on path from root (or common ancestors)
532. Find LCA in Binary Tree using RMQ
533. Maximum difference between node and its ancestor in Binary Tree
534. Print the path common to the two paths from the root to the two given nodes
535. Query for ancestor-descendant relationship in a tree
536. Iterative method to find ancestors of a given binary tree
537. Print Ancestors of a given node in Binary Tree
538. Kth ancestor of a node in binary tree
539. More >>
540. Misc :
541. Succinct Encoding of Binary Tree
542. Binary Indexed Tree : Range Updates and Point Queries
543. The Great Tree-List Recursion Problem
544. Custom Tree Problem
545. Tree Isomorphism Problem
546. Ways to color a skewed tree such that parent and child have different colors
547. Write a program to Delete a Tree
548. Delete leaf nodes with value as x
549. Non-recursive program to delete an entire binary tree
550. Write a program to Calculate Size of a tree
551. Iterative program to Calculate Size of a tree
552. Write a Program to Find the Maximum Depth or Height of a Tree
553. Iterative Method to find Height of Binary Tree
554. Height of a complete binary tree (or Heap) with N nodes
555. Height of binary tree considering even level leaves only
556. Find Height of Binary Tree represented by Parent array
557. How to determine if a binary tree is height-balanced?
558. Find height of a special binary tree whose leaf nodes are connected
559. Height of a generic tree from parent array
560. Diameter of a Binary Tree
561. Diameter of a Binary Tree in O(n) [A new method]
562. Possible edges of a tree for given diameter, height and vertices
563. Deepest right leaf node in a binary tree | Iterative approach
564. Sink Odd nodes in Binary Tree
565. Depth of the deepest odd level node in Binary Tree
566. Find depth of the deepest odd level leaf node
567. Find the Deepest Node in a Binary Tree
568. Deepest left leaf node in a binary tree | iterative approach
569. Deepest left leaf node in a binary tree
570. Find Minimum Depth of a Binary Tree
571. Replace node with depth in a binary tree
572. Maximum width of a binary tree
573. Vertical width of Binary tree | Set 1
574. Vertical width of Binary tree | Set 2
575. Find if given vertical level of binary tree is sorted or not
576. Check if a binary tree is sorted level-wise or not
577. Bottom View of a Binary Tree
578. Program to count leaf nodes in a binary tree
579. Iterative program to count leaf nodes in a Binary Tree
580. Count Non-Leaf nodes in a Binary Tree
581. Count half nodes in a Binary tree (Iterative and Recursive)
582. Count full nodes in a Binary tree (Iterative and Recursive)
583. Connect Nodes at same Level (Level Order Traversal)
584. Connect nodes at same level using constant extra space
585. Connect nodes at same level
586. Level with maximum number of nodes
587. Averages of Levels in Binary Tree
588. Largest value in each level of Binary Tree
589. Smallest value in each level of Binary Tree
590. Get Level of a node in a Binary Tree
591. Get level of a node in binary tree | iterative approach
592. Find mirror of a given node in Binary tree
593. Find largest subtree having identical left and right subtrees
594. Find Count of Single Valued Subtrees
595. Closest leaf to a given node in Binary Tree
596. Find the closest leaf in a Binary Tree
597. Iterative Search for a key ‘x’ in Binary Tree
598. Given a binary tree, how do you remove all the half nodes?
599. Swap Nodes in Binary tree of every k’th level
600. Pairwise Swap leaf nodes in a binary tree
601. Root to leaf paths having equal lengths in a Binary Tree
602. Root to leaf path with maximum distinct nodes
603. Maximum Consecutive Increasing Path Length in Binary Tree
604. Longest Path with Same Values in a Binary Tree
605. Remove nodes on root to leaf paths of length < K
606. Longest consecutive sequence in Binary tree
607. Path length having maximum number of bends
608. Number of turns to reach from one node to other in binary tree
609. Create loops of even and odd values in a binary tree
610. Find first non matching leaves in two binary trees
611. Get maximum left node in binary tree
612. Find a number in minimum steps
613. Factor Tree of a given Number
614. Number of full binary trees such that each node is product of its children
615. Number of subtrees having odd count of even numbers
616. Find distance from root to given node in a binary tree
617. Find distance between two given keys of a Binary Tree
618. Find right sibling of a binary tree with parent pointers
619. Find next right node of a given key
620. Tilt of Binary Tree
621. Find All Duplicate Subtrees
622. Top three elements in binary tree
623. Find maximum (or minimum) in Binary Tree
624. Extract Leaves of a Binary Tree in a Doubly Linked List
625. Minimum no. of iterations to pass information to all nodes in the tree
626. Binary Tree to Binary Search Tree Conversion
627. Sorted Linked List to Balanced BST
628. Sorted Array to Balanced BST
629. Transform a BST to greater sum tree
630. Construct all possible BSTs for keys 1 to N
631. Convert a BST to a Binary Tree such that sum of all greater keys is added to every key
632. BST to a Tree with sum of all smaller keys
633. In-place Convert BST into a Min-Heap
634. Convert BST to Min Heap
635. Construct BST from its given level order traversal
636. Reverse a path in BST using queue
637. Binary Tree to Binary Search Tree Conversion using STL set
638. Check given array of size n can represent BST of n levels or not
639. Convert a normal BST to Balanced BST
640. Merge Two Balanced Binary Search Trees
641. Merge two BSTs with limited extra space
642. Checking and Searching :
643. Find the node with minimum value in a Binary Search Tree
644. Check if the given array can represent Level Order Traversal of Binary Search Tree
645. Check if a given array can represent Preorder Traversal of Binary Search Tree
646. Lowest Common Ancestor in a Binary Search Tree
647. A program to check if a binary tree is BST or not
648. Find k-th smallest element in BST (Order Statistics in BST)
649. Check if each internal node of a BST has exactly one child
650. Check for Identical BSTs without building the trees
651. K’th Largest Element in BST when modification to BST is not allowed
652. K’th Largest element in BST using constant extra space
653. Second largest element in BST
654. K’th smallest element in BST using O(1) Extra Space
655. Check if given sorted sub-sequence exists in binary search tree
656. Simple Recursive solution to check whether BST contains dead end
657. Check if an array represents Inorder of Binary Search tree or not
658. Check if two BSTs contain same set of elements
659. Largest number in BST which is less than or equal to N
660. Maximum Unique Element in every subarray of size K
661. Iterative searching in Binary Search Tree
662. Find distance between two nodes of a Binary Search Tree
663. Count pairs from two BSTs whose sum is equal to a given value x
664. Find median of BST in O(n) time and O(1) space
665. Largest BST in a Binary Tree | Set 2
666. Remove BST keys outside the given range
667. Print BST keys in the given range
668. Print BST keys in given Range | O(1) Space
669. Count BST nodes that lie in a given range
670. Count BST subtrees that lie in given range
671. Remove all leaf nodes from the binary search tree
672. Sum of k smallest elements in BST
673. Inorder Successor in Binary Search Tree
674. Inorder predecessor and successor for a given key in BST
675. Inorder predecessor and successor for a given key in BST | Iterative Approach
676. Find if there is a triplet in a Balanced BST that adds to zero
677. Find a pair with given sum in a Balanced BST
678. Find a pair with given sum in BST
679. Maximum element between two nodes of BST
680. Find pairs with given sum such that pair elements lie in different BSTs
681. Find the closest element in Binary Search Tree
682. Find the largest BST subtree in a given Binary Tree
683. Replace every element with the least greater element on its right
684. Add all greater values to every node in a given BST
685. Red Black Tree and Threaded Binary Tree :
686. C Program for Red Black Tree Insertion
687. Left Leaning Red Black Tree (Insertion)
688. Threaded Binary Tree
689. Threaded Binary Tree | Insertion
690. Threaded Binary Search Tree | Deletion
691. Convert a Binary Tree to Threaded binary tree | Set 1 (Using Queue)
692. Convert a Binary Tree to Threaded binary tree | Set 2 (Efficient)
693. Threaded Binary Tree
694. Inorder Non-threaded Binary Tree Traversal without Recursion or Stack
695. Misc :
696. Sorted order printing of a given array that represents a BST
697. Two nodes of a BST are swapped, correct the BST
698. Floor and Ceil from a BST
699. Given n appointments, find all conflicting appointments
700. How to handle duplicates in Binary Search Tree?
701. Data Structure for a single resource reservations
702. How to implement decrease key or change key in Binary Search Tree?
703. Print Common Nodes in Two Binary Search Trees
704. Count inversions in an array | Set 2 (Using Self-Balancing BST)
705. Leaf nodes from Preorder of a Binary Search Tree
706. Leaf nodes from Preorder of a Binary Search Tree (Using Recursion)
707. Binary Search Tree insert with Parent Pointer
708. Minimum Possible value of |ai + aj – k| for given array and k.
709. Rank of an element in a stream
710. Special two digit numbers in a Binary Search Tree
711. Binary Heap
712. Time Complexity of building a heap
713. Applications of Heap Data Structure
714. Binomial Heap
715. Fibonacci Heap
716. Leftist Heap
717. K-ary Heap
718. Heap Sort
719. Iterative Heap Sort
720. K’th Largest Element in an array
721. K’th Smallest/Largest Element in Unsorted Array | Set 1
722. Sort an almost sorted array/
723. Tournament Tree (Winner Tree) and Binary Heap
724. Check if a given Binary Tree is Heap
725. How to check if a given array represents a Binary Heap?
726. Connect n ropes with minimum cost
727. Design an efficient data structure for given operations
728. Merge k sorted arrays | Set 1
729. Merge Sort Tree for Range Order Statistics
730. Sort numbers stored on different machines
731. Smallest Derangement of Sequence
732. Largest Derangement of a Sequence
733. K maximum sum combinations from two arrays
734. Maximum distinct elements after removing k elements
735. Maximum difference between two subsets of m elements
736. Height of a complete binary tree (or Heap) with N nodes
737. Heap Sort for decreasing order using min heap
738. Print all nodes less than a value x in a Min Heap.
739. Median of Stream of Running Integers using STL
740. Largest triplet product in a stream
741. Find k numbers with most occurrences in the given array
742. Convert BST to Min Heap
743. Merge two binary Max Heaps
744. K-th Largest Sum Contiguous Subarray
745. Minimum product of k integers in an array of positive Integers
746. Leaf starting point in a Binary Heap data structure
747. Why is Binary Heap Preferred over BST for Priority Queue?
748. Convert min Heap to max Heap
749. Given level order traversal of a Binary Tree, check if the Tree is a Min-Heap
750. Rearrange characters in a string such that no two adjacent are same
751. Implementation of Binomial Heap
752. Array Representation Of Binary Heap
753. Sum of all elements between k1’th and k2’th smallest elements
754. Minimum sum of two numbers formed from digits of an array
755. K’th largest element in a stream
756. k largest(or smallest) elements in an array | added Min Heap method
757. Median in a stream of integers (running integers)
758. Sort a nearly sorted (or K sorted) array
759. Graph and its representations
760. Breadth First Traversal for a Graph
761. Depth First Traversal for a Graph
762. Applications of Depth First Search
763. Applications of Breadth First Traversal
764. Graph representations using set and hash
765. Find Mother Vertex in a Graph
766. Transitive Closure of a Graph using DFS
767. Find K cores of an undirected Graph
768. Iterative Depth First Search
769. Count the number of nodes at given level in a tree using BFS
770. Count all possible paths between two vertices
771. Minimum initial vertices to traverse whole matrix with given conditions
772. Shortest path to reach one prime to other by changing single digit at a time
773. Water Jug problem using BFS
774. Count number of trees in a forest
775. BFS using vectors & queue as per the algorithm of CLRS
776. Level of Each node in a Tree from source node
777. Construct binary palindrome by repeated appending and trimming
778. Transpose graph
779. Path in a Rectangle with Circles
780. Height of a generic tree from parent array
781. BFS using STL for competitive coding
782. DFS for a n-ary tree (acyclic graph) represented as adjacency list
783. Maximum number of edges to be added to a tree so that it stays a Bipartite graph
784. A Peterson Graph Problem
785. Implementation of Graph in JavaScript
786. Print all paths from a given source to a destination using BFS
787. Minimum number of edges between two vertices of a Graph
788. Count nodes within K-distance from all nodes in a set
789. Bidirectional Search
790. Minimum edge reversals to make a root
791. BFS for Disconnected Graph
792. Move weighting scale alternate under given constraints
793. Best First Search (Informed Search)
794. Number of pair of positions in matrix which are not accessible
795. Maximum product of two non-intersecting paths in a tree
796. Delete Edge to minimize subtree sum difference
797. Find the minimum number of moves needed to move from one cell of matrix to another
798. Minimum steps to reach target by a Knight | Set 1
799. Minimum number of operation required to convert number x into y
800. Minimum steps to reach end of array under constraints
801. Find the smallest binary digit multiple of given number
802. Roots of a tree which give minimum height
803. Stepping Numbers
804. Clone an Undirected Graph
805. Sum of the minimum elements in all connected components of an undirected graph
806. Check if two nodes are on same path in a tree
807. A matrix probability question
808. Find length of the largest region in Boolean Matrix
809. Iterative Deepening Search(IDS) or Iterative Deepening Depth First Search(IDDFS)
810. Graph Cycle :
811. Detect Cycle in a Directed Graph
812. Detect cycle in an undirected graph
813. Detect cycle in a direct graph using colors
814. Assign directions to edges so that the directed graph remains acyclic
815. Detect a negative cycle in a Graph | (Bellman Ford)
816. Cycles of length n in an undirected and connected graph
817. Detecting negative cycle using Floyd Warshall
818. Check if there is a cycle with odd weight sum in an undirected graph
819. Check if a graphs has a cycle of odd length
820. Clone a Directed Acyclic Graph
821. Check loop in array according to given constraints
822. Disjoint Set (Or Union-Find) | Set 1
823. Union-Find Algorithm | Set 2
824. Union-Find Algorithm | (Union By Rank and Find by Optimized Path Compression)
825. Magical Indices in an array
826. Topological Sorting :
827. Topological Sorting
828. All topological sorts of a Directed Acyclic Graph
829. Kahn’s Algorithm for Topological Sorting
830. Maximum edges that can be added to DAG so that is remains DAG
831. Longest path between any pair of vertices
832. Longest Path in a Directed Acyclic Graph
833. Longest Path in a Directed Acyclic Graph | Set 2
834. Topological Sort of a graph using departure time of vertex
835. Given a sorted dictionary of an alien language, find order of characters
836. Minimum Spanning Tree :
837. Prim’s Minimum Spanning Tree (MST))
838. Applications of Minimum Spanning Tree Problem
839. Prim’s MST for Adjacency List Representation
840. Kruskal’s Minimum Spanning Tree Algorithm
841. Boruvka’s algorithm for Minimum Spanning Tree
842. Minimum cost to connect all cities
843. Steiner Tree
844. Reverse Delete Algorithm for Minimum Spanning Tree
845. Total number of Spanning Trees in a Graph
846. Minimum Product Spanning Tree
847. BackTracking :
848. Find if there is a path of more than k length from a source
849. Tug of War
850. The Knight-Tour Problem
851. Rat in a Maze
852. n-Queen’s Problem
853. m Coloring Problem
854. Hamiltonian Cycle
855. Permutation of numbers such that sum of two consecutive numbers is a perfect square
856. Shortest Paths :
857. Dijkstra’s shortest path algorithm
858. Dijkstra’s Algorithm for Adjacency List Representation
859. Bellman–Ford Algorithm
860. Floyd Warshall Algorithm
861. Johnson’s algorithm for All-pairs shortest paths
862. Shortest Path in Directed Acyclic Graph
863. Shortest path with exactly k edges in a directed and weighted graph
864. Dial’s Algorithm
865. Printing paths in Dijsktra’s Algorithm
866. Shortest path of a weighted graph where weight is 1 or 2
867. Multistage Graph (Shortest Path)
868. Shortest path in an unweighted graph
869. Minimize the number of weakly connected nodes
870. Betweenness Centrality (Centrality Measure)
871. Comparison of Dijkstra’s and Floyd–Warshall algorithms
872. Karp’s minimum mean (or average) weight cycle algorithm
     1. BFS (Shortest Path in a Binary Weight Graph)
873. Find minimum weight cycle in an undirected graph
874. Minimum Cost Path with Left, Right, Bottom and Up moves allowed
875. Minimum edges to reverse to make path from a src to a dest
876. Find Shortest distance from a guard in a Bank
877. Connectivity :
878. Find if there is a path between two vertices in a directed graph
879. Connectivity in a directed graph
880. Articulation Points (or Cut Vertices) in a Graph
881. Biconnected Components
882. Biconnected graph
883. Bridges in a graph
884. Eulerian path and circuit
885. Fleury’s Algorithm for printing Eulerian Path or Circuit
886. Strongly Connected Components
887. Transitive closure of a graph
888. Find the number of islands
889. Find the number of Islands | Set 2 (Using Disjoint Set)
890. Count all possible walks from a source to a destination with exactly k edges
891. Euler Circuit in a Directed Graph
892. Count the number of non-reachable nodes
893. Find the Degree of a Particular vertex in a Graph
894. Check if a given graph is tree or not
895. Minimum edges required to add to make Euler Circuit
896. Eulerian Path in undirected graph
897. Find if there is a path of more than k length
898. Length of shortest chain to reach the target word
899. Print all paths from a given source to destination
900. Find minimum cost to reach destination using train
901. Find if an array of strings can be chained to form a circle | Set 1
902. Find if an array of strings can be chained to form a circle | Set 2
903. Tarjan’s Algorithm to find strongly connected Components
904. Number of loops of size k starting from a specific node
905. Paths to travel each nodes using each edge (Seven Bridges of Königsberg)
906. Number of cyclic elements in an array where we can jump according to value
907. Number of groups formed in a graph of friends
908. Minimum cost to connect weighted nodes represented as array
909. Count single node isolated sub-graphs in a disconnected graph
910. Calculate number of nodes between two vertices in an acyclic Graph by Disjoint Union method
911. Dynamic Connectivity | Set 1 (Incremental)
912. Check if a graph is strongly connected | Set 1 (Kosaraju using DFS)
913. Check if a given directed graph is strongly connected | Set 2 (Kosaraju using BFS)
914. Check if removing a given edge disconnects a graph
915. Find all reachable nodes from every node present in a given set
916. Connected Components in an undirected graph
917. k’th heaviest adjacent node in a graph where each vertex has weight
918. Maximum Flow :
919. Ford-Fulkerson Algorithm for Maximum Flow Problem
920. Find maximum number of edge disjoint paths between two vertices
921. Find minimum s-t cut in a flow network
922. Maximum Bipartite Matching
923. Channel Assignment Problem
924. Push Relabel- Set 1-Introduction
925. Push Relabel- Set 2- Implementation
926. Karger’s Algorithm- Set 1- Introduction and Implementation
927. Karger’s Algorithm- Set 2 – Analysis and Applications
928. Dinic’s algorithm for Maximum Flow
929. Max Flow Problem Introduction
930. STL Implementation of Algorithms :
931. Kruskal’s Minimum Spanning Tree using STL in C++
932. Prim’s Algorithm using Priority Queue STL
933. Dijkstra’s Shortest Path Algorithm using STL
934. Dijkstra’s Shortest Path Algorithm using set in STL
935. Graph implementation using STL for competitive programming | Set 2 (Weighted graph)
936. Hard Problems :
937. Graph Coloring (Introduction and Applications)
938. Greedy Algorithm for Graph Coloring
939. Traveling Salesman Problem (TSP) Implementation
940. Travelling Salesman Problem (Naive and Dynamic Programming)
941. Travelling Salesman Problem (Approximate using MST)
942. Vertex Cover Problem | Set 1 (Introduction and Approximate Algorithm)
943. K Centers Problem | Set 1 (Greedy Approximate Algorithm)
944. Erdos Renyl Model (for generating Random Graphs)
945. Clustering Coefficient in Graph Theory
946. Chinese Postman or Route Inspection | Set 1 (introduction)
947. Hierholzer’s Algorithm for directed graph
948. Misc :
949. Number of triangles in an undirected Graph
950. Number of triangles in directed and undirected Graph
951. Check whether a given graph is Bipartite or not
952. Snake and Ladder Problem
953. Minimize Cash Flow among a given set of friends who have borrowed money from each other
954. Boggle (Find all possible words in a board of characters)
955. Hopcroft Karp Algorithm for Maximum Matching-Introduction
956. Hopcroft Karp Algorithm for Maximum Matching-Implementation
957. Minimum Time to rot all oranges
958. Find same contents in a list of contacts
959. Hypercube Graph
960. Check for star graph
961. Optimal read list for a given number of days
962. Print all jumping numbers smaller than or equal to a given value
963. Fibonacci Cube Graph
964. Barabasi Albert Graph (for Scale Free Models)
965. Construct a graph from given degrees of all vertices
966. Degree Centrality (Centrality Measure)
967. Katz Centrality (Centrality Measure)
968. Mathematics | Graph theory practice questions
969. 2-Satisfiability (2-SAT) Problem
970. Determine whether a universal sink exists in a directed graph
971. Number of sink nodes in a graph
972. Largest subset of Graph vertices with edges of 2 or more colors
973. NetworkX : Python software package for study of complex networks
974. Generate a graph using Dictionary in Python
975. Count number of edges in an undirected graph
976. Two Clique Problem (Check if Graph can be divided in two Cliques)
977. Check whether given degrees of vertices represent a Graph or Tree
978. Finding minimum vertex cover size of a graph using binary search
979. Stable Marriage Problem
980. Sum of dependencies in a graph
981. Print a Binary Tree in Vertical Order | Set 2 (Hashmap based Method)
982. Find whether an array is subset of another array
983. Union and Intersection of two linked lists
984. Given an array A[] and a number x, check for pair in A[] with sum as x
985. Minimum delete operations to make all elements of array same
986. Minimum operation to make all elements equal in array
987. Maximum distance between two occurrences of same element in array
988. Count maximum points on same line
989. Check if a given array contains duplicate elements within k distance from each other
990. Find duplicates in a given array when elements are not limited to a range
991. Find top k (or most frequent) numbers in a stream
992. Most frequent element in an array
993. Smallest subarray with all occurrences of a most frequent element
994. First element occurring k times in an array
995. Given an array of pairs, find all symmetric pairs in it
996. Find the only repetitive element between 1 to n-1
997. Find any one of the multiple repeating elements in read only array
998. Find top three repeated in array
999. Group multiple occurrence of array elements ordered by first occurrence
1000. How to check if two given sets are disjoint?
1001. Non-overlapping sum of two sets
1002. Find elements which are present in first array and not in second
1003. Check if two arrays are equal or not
1004. Pair with given sum and maximum shortest distance from end
1005. Pair with given product | Set 1 (Find if any pair exists)
1006. Find missing elements of a range
1007. k-th missing element in increasing sequence which is not present in a given sequence
1008. Find pair with greatest product in array
1009. Minimum number of subsets with distinct elements
1010. Remove minimum number of elements such that no common element exist in both arra
1011. Count items common to both the lists but with different prices
1012. Minimum Index Sum for Common Elements of Two Lists
1013. Find pairs with given sum such that elements of pair are in different rows
1014. Common elements in all rows of a given matrix
1015. Find distinct elements common to all rows of a matrix
1016. Find all permuted rows of a given row in a matrix
1017. Change the array into a permutation of numbers from 1 to n
1018. Count pairs with given sum
1019. Count pairs from two sorted arrays whose sum is equal to a given value x
1020. Count pairs from two linked lists whose sum is equal to a given value
1021. Count quadruples from four sorted arrays whose sum is equal to a given value x
1022. Number of subarrays having sum exactly equal to k
1023. Count pairs whose products exist in array
1024. Given two unsorted arrays, find all pairs whose sum is x
1025. Frequency of each element in an unsorted array
1026. Sort elements by frequency
1027. Find pairs in array whose sums already exist in array
1028. Find all pairs (a, b) in an array such that a % b = k
1029. Convert an array to reduced form | Set 1 (Simple and Hashing)
1030. Return maximum occurring character in an input string
1031. Group words with same set of characters
1032. Second most repeated word in a sequence
1033. Smallest element repeated exactly ‘k’ times (not limited to small range)
1034. Numbers with prime frequencies greater than or equal to k
1035. Find k numbers with most occurrences in the given array
1036. Find the first repeating element in an array of integers
1037. Find sum of non-repeating (distinct) elements in an array
1038. Non-Repeating Element
1039. k-th distinct (or non-repeating) element in an array.
1040. Print All Distinct Elements of a given integer array
1041. Only integer with positive value in positive negative value in array
1042. Pairs of Positive Negative values in an array
1043. Intermediate :
1044. Find Itinerary from a given list of tickets
1045. Find number of Employees Under every Employee
1046. Count divisible pairs in an array
1047. Check if an array can be divided into pairs whose sum is divisible by k
1048. Longest subarray with sum divisible by k
1049. Subarray with no pair sum divisible by K
1050. Print array elements that are divisible by at-least one other
1051. Find three element from different three arrays such that that a + b + c = sum
1052. Find four elements a, b, c and d in an array such that a+b = c+d
1053. Find the largest subarray with 0 sum
1054. Printing longest Increasing consecutive subsequence
1055. Longest Increasing consecutive subsequence
1056. Longest subsequence such that difference between adjacents is one | Set 2
1057. Longest Consecutive Subsequence
1058. Largest increasing subsequence of consecutive integers
1059. Count subsets having distinct even numbers
1060. Count distinct elements in every window of size k
1061. Maximum possible sum of a window in an array such that elements of same window in other array are unique
1062. Distributing items when a person cannot take more than two items of same type
1063. Design a data structure that supports insert, delete, search and getRandom in constant time
1064. Check if array contains contiguous integers with duplicates allowed
1065. Length of the largest subarray with contiguous elements
1066. Find if there is a subarray with 0 sum
1067. Print all subarrays with 0 sum
1068. Find subarray with given sum | Set 2 (Handles Negative Numbers)
1069. Find four elements that sum to a given value
1070. Implementing our Own Hash Table with Separate Chaining in Java
1071. Implementing own Hash Table with Open Addressing Linear Probing in C++
1072. Vertical Sum in a given Binary Tree
1073. Group Shifted String
1074. Minimum insertions to form a palindrome with permutations allowed
1075. Check for Palindrome after every character replacement Query
1076. Maximum length subsequence with difference between adjacent elements as either 0 or 1 | Set 2
1077. Maximum difference between frequency of two elements such that element having greater frequency is also greater
1078. Difference between highest and least frequencies in an array
1079. Maximum difference between first and last indexes of an element in array
1080. Maximum possible difference of two subsets of an array
1081. Sorting using trivial hash function
1082. Smallest subarray with k distinct numbers
1083. Longest subarray not having more then K distinct elements
1084. Sum of f(a[i], a[j]) over all pairs in an array of n integers
1085. Find number of pairs in an array such that their XOR is 0
1086. Maximize elements using another array
1087. Hard :
1088. Clone a Binary Tree with Random Pointers
1089. Largest subarray with equal number of 0s and 1s
1090. Longest subarray having count of 1s one more than count of 0s
1091. Longest subarray having count of 1s one more than count of 0s
1092. Count Substrings with equal number of 0s, 1s and 2s
1093. Print all triplets in sorted array that form AP
1094. All unique triplets that sum up to a given value
1095. Find all triplets with zero sum
1096. Count number of triplets with product equal to given number
1097. Count of index pairs with equal elements in an array
1098. Palindrome Substring Queries
1099. Find smallest range containing elements from k lists
1100. Range Queries for Frequencies of array elements
1101. Elements to be added so that all elements of a range are present in array
1102. Cuckoo Hashing – Worst case O(1) Lookup!
1103. Subarrays with distinct elements
1104. Count subarrays having total distinct elements same as original array
1105. Count subarrays with same even and odd elements
1106. Minimum number of distinct elements after removing m items
1107. Distributing items when a person cannot take more than two items of same type
1108. Maximum consecutive numbers present in an array
1109. Maximum array from two given arrays keeping order same
1110. Maximum number of chocolates to be distributed equally among k students
1111. Find largest d in array such that a + b + c = d
1112. Find Sum of all unique sub-array sum for a given array.
1113. Misc :
1114. Advantages of BST over Hash Table
1115. Internal Working of HashMap in Java
1116. Hash Table vs STL Map
1117. Recaman’s sequence
1118. C++ program for hashing with chaining
1119. Largest subset whose all elements are Fibonacci numbers
1120. Pairs of Amicable Numbers
1121. Find All Duplicate Subtrees
1122. Hash Table vs STL Map
1123. Find if there is a rectangle in binary matrix with corners as 1
1124. Maximum area rectangle by picking four sides from array
1125. Root to leaf path with maximum distinct nodes
1126. Game of replacing array elements
1127. Length of longest strict bitonic subsequence
1128. Last seen array element (last appearance is earliest)