**Linked List:**

|  |  |  |
| --- | --- | --- |
| ID | Problem | Status |
|  | Introduction to Linked List |  |
|  | Linked List vs Array |  |
|  | Linked List Insertion |  |
|  | Linked List Deletion (Deleting a given key) |  |
|  | Linked List Deletion (Deleting a key at given position) |  |
|  | Find Length of a Linked List (Iterative and Recursive) |  |
|  | Search an element in a Linked List (Iterative and Recursive) |  |
|  | Swap nodes in a linked list without swapping data |  |
|  | Write a function to get Nth node in a Linked List |  |
|  | Print the middle of a given linked list |  |
|  | Nth node from the end of a Linked List |  |
|  | Write a function to delete a Linked List |  |
|  | Write a function that counts the number of times a given int occurs in a Linked List |  |
|  | Reverse a linked list |  |
|  | Detect loop in a linked list |  |
|  | Merge two sorted linked lists |  |
|  | Generic Linked List in C |  |
|  | Function to check if a singly linked list is palindrome |  |
|  | Intersection point of two Linked Lists. |  |
|  | Recursive function to print reverse of a Linked List |  |
|  | Remove duplicates from a sorted linked list |  |
|  | Remove duplicates from an unsorted linked list |  |
|  | Pairwise swap elements of a given linked list |  |
|  | Move last element to front of a given Linked List |  |
|  | Intersection of two Sorted Linked Lists |  |
|  | Delete alternate nodes of a Linked List |  |
|  | Alternating split of a given Singly Linked List |  |
|  | Identical Linked Lists |  |
|  | Merge Sort for Linked Lists |  |
|  | Reverse a Linked List in groups of given size |  |
|  | Reverse alternate K nodes in a Singly Linked List |  |
|  | Delete nodes which have a greater value on right side |  |
|  | Segregate even and odd nodes in a Linked List |  |
|  | Detect and Remove Loop in a Linked List |  |
|  | Add two numbers represented by linked lists | Set 1 |  |
|  | Delete a given node in Linked List under given constraints |  |
|  | Union and Intersection of two Linked Lists |  |
|  | Find a triplet from three linked lists with sum equal to a given number |  |
|  | Rotate a Linked List |  |
|  | Flattening a Linked List |  |
|  | Add two numbers represented by linked lists | Set 2 |  |
|  | Sort a linked list of 0s, 1s and 2s |  |
|  | Flatten a multilevel linked list |  |
|  | Delete N nodes after M nodes of a linked list |  |
|  | QuickSort on Singly Linked List |  |
|  | Merge a linked list into another linked list at alternate positions |  |
|  | Pairwise swap elements of a given linked list by changing links |  |
|  | Given a linked list of line segments, remove middle points |  |
|  | Clone a linked list with next and random pointer | Set 1 |  |
|  | Clone a linked list with next and random pointer | Set 2 |  |
|  | Insertion Sort for Singly Linked List |  |
|  | Point to next higher value node in a linked list with an arbitrary pointer |  |
|  | Rearrange a given linked list in-place. |  |
|  | Sort a linked list that is sorted alternating ascending and descending orders. |  |
|  | Select a Random Node from a Singly Linked List |  |
|  | Merge two sorted linked lists such that merged list is in reverse order |  |
|  | Compare two strings represented as linked lists |  |
|  | Rearrange a linked list such that all even and odd positioned nodes are together |  |
|  | Rearrange a Linked List in Zig-Zag fashion |  |
|  | Add 1 to a number represented as linked list |  |
|  | Point arbit pointer to greatest value right side node in a linked list |  |
|  | Merge two sorted linked lists such that merged list is in reverse order |  |
|  | Check if a linked list of strings forms a palindrome |  |
|  | Sort linked list which is already sorted on absolute values |  |
|  | Delete last occurrence of an item from linked list |  |
|  | Delete a Linked List node at a given position |  |
|  | Linked List in java |  |
|  | In-place Merge two linked lists without changing links of first list |  |
|  | Delete middle of linked list |  |
|  | Merge K sorted linked lists | Set 1 |  |
|  | Decimal Equivalent of Binary Linked List |  |
|  | Flatten a multi-level linked list | Set 2 (Depth wise) |  |
|  | Rearrange a given list such that it consists of alternating minimum maximum elements |  |
|  | Subtract Two Numbers represented as Linked Lists |  |
|  | Find pair for given sum in a sorted singly linked without extra space |  |
|  | Iteratively Reverse a linked list using only 2 pointers (An Interesting Method) |  |
|  | Partitioning a linked list around a given value and keeping the original order |  |
|  | Check linked list with a loop is palindrome or not |  |
|  | Clone a linked list with next and random pointer in O(1) space |  |
|  | Length of longest palindrome list in a linked list using O(1) extra space |  |
|  | Adding two polynomials using Linked List |  |
|  | Implementing Iterator pattern of a single Linked List |  |
|  | Move all occurrences of an element to end in a linked list |  |
|  | Remove all occurrences of duplicates from a sorted Linked List |  |
|  | Remove every k-th node of the linked list |  |
|  | Check whether the length of given linked list is Even or Odd |  |
|  | Union and Intersection of two linked lists | Set-2 (Using Merge Sort) |  |
|  | Multiply two numbers represented by Linked Lists |  |
|  | Union and Intersection of two linked lists | Set-3 (Hashing) |  |
|  | Find the sum of last n nodes of the given Linked List |  |
|  | Count pairs from two linked lists whose sum is equal to a given value |  |
|  | Merge k sorted linked lists | Set 2 (Using Min Heap) |  |
|  | Recursive selection sort for singly linked list | Swapping node links |  |
|  | Find length of loop in linked list |  |
|  | Reverse a Linked List in groups of given size | Set 2 |  |
|  | Insert node into the middle of the linked list |  |
|  | Merge two sorted lists (in-place) |  |
|  | Sort a linked list of 0s, 1s and 2s by changing links |  |
|  | Insert a node after the n-th node from the end |  |
|  | Rotate Linked List block wise |  |
|  | Count rotations in sorted and rotated linked list |  |
|  | Make middle node head in a linked list |  |
|  | Circular Linked List Introduction and Applications, |  |
|  | Circular Linked List Traversal |  |
|  | Split a Circular Linked List into two halves |  |
|  | Sorted insert for circular linked list |  |
|  | Check if a linked list is Circular Linked List |  |
|  | Convert a Binary Tree to a Circular Doubly Link List |  |
|  | Circular Singly Linked List | Insertion |  |
|  | Deletion from a Circular Linked List |  |
|  | Circular Queue | Set 2 (Circular Linked List Implementation) |  |
|  | Count nodes in Circular linked list |  |
|  | Josephus Circle using circular linked list |  |
|  | Convert singly linked list into circular linked list |  |
|  | Doubly Linked List Introduction and Insertion |  |
|  | Delete a node in a Doubly Linked List |  |
|  | Reverse a Doubly Linked List |  |
|  | The Great Tree-List Recursion Problem. |  |
|  | Copy a linked list with next and arbit pointer |  |
|  | QuickSort on Doubly Linked List |  |
|  | Swap Kth node from beginning with Kth node from end in a Linked List |  |
|  | Merge Sort for Doubly Linked List |  |
|  | Create a Doubly Linked List from a Ternary Tree |  |
|  | Find pairs with given sum in doubly linked list |  |
|  | Insert value in sorted way in a sorted doubly linked list |  |
|  | Delete a Doubly Linked List node at a given position |  |
|  | Count triplets in a sorted doubly linked list whose sum is equal to a given value x |  |
|  | Remove duplicates from a sorted doubly linked list |  |
|  | Delete all occurrences of a given key in a doubly linked list |  |
|  | Remove duplicates from an unsorted doubly linked list |  |
|  | Sort the biotonic doubly linked list |  |
|  | Sort a k sorted doubly linked list |  |
|  | Convert a given Binary Tree to Doubly Linked List |  |
|  | Merge Two Balanced Binary Search Trees |  |
|  | In-place conversion of Sorted DLL to Balanced BST |  |
|  | XOR Linked List – A Memory Efficient Doubly Linked List | Set 1 |  |
|  | Skip List | Set 1 (Introduction) |  |
|  | Skip List | Set 2 (Insertion) |  |
|  | Skip List | Set 3 (Searching and Deletion) |  |
|  | Reverse a stack without using extra space in O(n) |  |
|  | An interesting method to print reverse of a linked list |  |
|  | Linked List representation of Disjoint Set Data Structures |  |
|  | Sublist Search (Search a linked list in another list) |  |
|  | Doubly Circular Linked List | Set 1 (Introduction and Insertion) |  |
|  | Doubly Circular Linked List | Set 2 (Deletion) |  |
|  | How to insert elements in C++ STL List ? |  |
|  | Unrolled Linked List | Set 1 (Introduction) |  |
|  | A Programmer’s approach of looking at Array vs. Linked List |  |
|  | How to write C functions that modify head pointer of a Linked List? |  |
|  | Given a linked list which is sorted, how will you insert in sorted way |  |
|  | Can we reverse a linked list in less than O(n)? |  |
|  | Practice questions for Linked List and Recursion |  |
|  | Construct a Maximum Sum Linked List out of two Sorted Linked Lists having some Common nodes |  |
|  | Given only a pointer to a node to be deleted in a singly linked list, how do you delete it? |  |
|  | Why Quick Sort preferred for Arrays and Merge Sort for Linked Lists? |  |
|  | Squareroot(n)-th node in a Linked List |  |
|  | Find the fractional (or n/k – th) node in linked list |  |
|  | Find modular node in a linked list |  |
|  | Construct a linked list from 2D matrix |  |
|  | Find smallest and largest elements in singly linked list |  |
|  | Arrange consonants and vowels nodes in a linked list |  |
|  | Partitioning a linked list around a given value and If we don’t care about making the elements of the list “stable” |  |
|  | Modify contents of Linked List |  |
|  | The Great Tree-List Recursion Problem |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |