

DSA Topics

LinkedList

All basic operations like insertion, deletion, traversal, search, find, etc.

Insert a node in a sorted linked list.

1. Reverse linked list (iterative & recursive)

2. Reverse linked list in a group of given size k (with & w/o recursion)

For a given K value (K > 0) reverse blocks of K nodes in a list.

Example: Input: 1 2 3 4 5 6 7 8 9 10. Output for different K values:

For K = 2: 2 1 4 3 6 5 8 7 10 9

For K = 3: 3 2 1 6 5 4 9 8 7 10

For K = 4: 4 3 2 1 8 7 6 5 9 10

Reverse the linked list in pairs. If you have a linked list that holds 1 → 2 → 3 → 4 → X, then after the function has been called the linked list would hold 2 → 1 → 4 → 3 → X.

3. Detect and delete loop in LL, isloop/iscycle, issorted /

Check whether the given linked list is either NULL-terminated or ends in a cycle (cyclic)., Floyd cycle finding algorithm.. If there is a cycle, find the length of the loop.

OR

Are given a pointer to the first element of a linked list L. There are two possibilities for L: it either ends (snake) or its last element points back to one of the earlier elements in the list (snail). Give an algorithm that tests whether a given list L is a snake or a snail.

4. Find starting point/node of loop

5. Remove duplicates in unsorted, sorted LL, combine duplicates in LL, Doubly LL

6. Write a Program to Move the last element to Front in a Linked List.

7. Add "1" to a number represented as a Linked List.

8. Add two numbers represented by linked lists.

9. Intersection of two Sorted Linked List.

Union of two linked lists.

10. Intersection Point of two Linked Lists.

11. Merge Sort For Linked lists., merge k sorted lists. [Very Important]

Merge two sorted linked lists such that merged list is in reverse order.

12. Quicksort for Linked Lists. [Very Important]

13. Find the middle Element of a linked list. & delete middle of the LL

14. Check if a linked list is a circular linked list.

15. Split a Circular linked list into two halves. (without counting no of nodes in LL)

Split a Circular Linked List into two equal parts. If the number of nodes in the list are odd then make first list one node extra than second list.

16. Write a Program to check whether the Singly Linked list is a palindrome or not.
(Check if the LL of strings form a palindrome)/
Function to check if a singly linked list is palindrome.

17. Deletion from a Circular Linked List.

Given only a pointer/reference to a node to be deleted in a singly list.

18. Reverse a Doubly Linked list.

Given a binary tree convert it to doubly linked list.

19. Find pairs with a given sum in a DLL.,count pairs whose sum is equal to x.

Pairwise swap elements of a given linked list.

20. Count triplets in a sorted DLL whose sum is equal to given value "X".

21. Sort a "k"sorted Doubly Linked list.[Very IMP]

22. Rotate DoublyLinked list by N nodes.

23. Rotate a Doubly Linked list in group of Given Size.[Very IMP]

24. Flatten a Linked List

25. Sort a LL of 0's, 1's and 2's

26. Clone a linked list with next and random pointer.
Given a linked list consists of data, a next pointer and also a random pointer
which points to a random node of the list. Give an algorithm for cloning the list.

27. Merge K sorted Linked list.

Given two sorted Linked Lists, how to merge them into the third list in sorted order?

28. Multiply 2 no. represented by LL

29. Delete nodes which have a greater value on right side

30. Segregate even and odd nodes in a Linked List

31. Program for n'th node from the end of a Linked List

32. Find the first non-repeating character from a stream of characters

33. Polynomials operations using LL:Add,multiply,expressions
(coefficient,exponent,pointer)
LL implementation of polynomial ADT

34. Count and sum of node elements in LL

35. Doubly linked list operations : Insertion & deletion

36. Circular linked list operations : Insertion and deletion

Check if a linked list is Circular Linked List.

Sorted insert for circular linked list.

37. Realizing linkedlist using arrays and pointers
38. Max and min element node in LL
39. String operations and reversal using LL and pointers
40. Implementation of stacks and queues.
41. Implementation of graphs: Adjacency list representation of graphs is most popular which uses linked list to store adjacent vertices.
42. Dynamic memory allocation: We use linked list of free blocks.
43. Maintaining directory of names. Performing arithmetic operations on long integers
44. Manipulation of polynomials by storing constants in the node of linked list. Representing sparse matrices using LL(Sparse matrix representations using arrays & LL), add 2 poly rep as LL
45. Circular Doubly Linked Lists are used for implementation of advanced data structures like Fibonacci Heap.
46. Remove nth node from the end of the list.: Use two pointers pNthNode and pTemp.
Remove every k-th node of the linked list.
47. Length of longest palindrome list in a linked list using O(1) extra space.
48. Round robin algo using circular LL.
49. Counting Nodes in a Circular Linked List
50. Printing the Contents of a Circular Linked List.
51. 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. List1 may have n nodes before it reaches the intersection point, and List2 might have m nodes before it reaches the intersection point where m and n may be $m = n$, $m < n$ or $m > n$. Give an algorithm for finding the merging point.
Given two singly linked lists of size N and M, write a program to get the point where two linked lists intersect each other.(INTERSECTION POINT IN Y SHAPED LL)
(FIND INTERSECTING NODE)
Also use stacks to solve the scenario.
52. How will you display a Linked List from the end?
53. Check whether the given Linked List length is even or odd?
54. If the head of a Linked List is pointing to kth element, then how will you get the

elements before kth element?
(Memory Efficient Linked Lists [XOR Linked Lists])

55. If we want to concatenate two linked lists which of the following gives $O(1)$ complexity?

- 1) Singly linked lists
- 2) Doubly linked lists
- 3) Circular doubly linked lists

56. Josephus Circle: N people have decided to elect a leader by arranging themselves in a circle and eliminating every M th person around the circle, closing ranks as each person drops out. Find which person will be the last one remaining (with rank 1).

57. Given a linked list with even and odd numbers, create an algorithm for making changes to the list in such a way that all even numbers appear at the beginning.

58. Find modular node: Given a singly linked list, write a function to find the last element from the beginning whose $n \% k == 0$, where n is the number of elements in the list and k is an integer constant. For example, if $n = 19$ and $k = 3$ then we should return 18th node.

59. 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 16th node.

60. Find fractional node: Given a singly linked list, write a function to find the (n/k) th element, where n is the number of elements in the list.

61. Find sq.root nth node: Given a singly linked list, write a function to find the square root nth element, where n is the number of elements in the list. Assume the value of n is not known in advance.

62. Given two lists List 1 = $\{A_1, A_2, \dots, A_n\}$ and List2 = $\{B_1, B_2, \dots, B_m\}$ with data (both lists) in ascending order. Merge them into the third list in ascending order so that the merged list will be:

63. Median in an infinite series of integers

64. Given a linked list, how do you modify it such that all the even numbers appear before all the odd numbers in the modified linked list?

65. Given two linked lists, each list node with one integer digit, add these two linked lists. The result should be stored in the third linked list. Also note that the head node contains the most significant digit of the number.

66. Given a list, List1 = $\{A_1, A_2, \dots, A_{n-1}; A_n\}$ with data, reorder it to $\{A_1, A_n, A_2, A_{n-1}\}$ without using any extra space.(Reorder list)

67. Given two sorted linked lists, given an algorithm for the printing common elements of them.

68. Farey fractions of level one are defined as sequence 1 0

1, 1/2. This sequence is extended

in level two to form a sequence 1

0

1, 1/2

2, 1/2

1/2, sequence 1

0

1, 1/2

3, 1/2

2, 1/2

3, 1/2

1/2 at level three, sequence

1

0

1, 1/2

4, 1/2

3, 1/2

2, 1/2

3, 1/2

4, 1/2

1/2 at level four, so that at each level n, a new fraction a/b

c/d is inserted

between two neighbor fractions a

c and b

d only if $c + d \leq n$. Write a program that for a

number n entered by the user creates-by constantly extending it-a linked list of fractions at level n and then displays them(Drozdek pg 145)

69. Write a simple airline ticket reservation program. The program should display a menu with the following options: reserve a ticket, cancel a reservation, check whether a ticket is reserved for a particular person, and display the passengers. The

information is maintained on an alphabetized linked list of names. In a simpler version of the program, assume that tickets are reserved for only one flight. In a fuller version, place no limit on the number of flights. Create a linked list of flights

with each node including a pointer to a linked list of passengers.

70. Add two numbers represented by LL

71. Add 1 to a number represented as LL

72. Detect nodes having greater value on the right

73. first non repeating character in a stream

74. Given a Singly Linked List of size N, delete all alternate nodes of the list

75. Given a singly linked list and a key, count the number of occurrences of given key in the linked list.

Example 1:

Input:

N = 7

Link List = 1->2->1->2->1->3->1

search_for = 1

Output: 4

Explanation: 1 appears 4 times.

76. Delete N nodes after M nodes of a linked list

EasyAccuracy: 37.19%Submissions: 41584Points: 2

Given a linked list, delete N nodes after skipping M nodes of a linked list until

the last of the linked list.

Example:

Input:

```
2
8
2 1
9 1 3 5 9 4 10 1
6
6 1
1 2 3 4 5 6
```

Output:

```
9 1 5 9 10 1
1 2 3 4 5 6
```

Explanation:

Deleting one node after skipping the M nodes each time, we have list as 9-> 1-> 5-> 9-> 10-> 1.

77. Swap kth nodes from the ends.

78. Given a singly linked list, the task is to rearrange it in a way that all odd position nodes are together and all even positions node are together. Assume the first element to be at position 1 followed by second element at position 2 and so on.

Note: You should place all odd positioned nodes first and then the even positioned ones. (considering 1 based indexing). Also, the relative order of odd positioned nodes and even positioned nodes should be maintained.

Example 1:

Input:

LinkedList: 1->2->3->4

Output: 1 3 2 4

Explanation:

Odd elements are 1, 3 and even elements are 2, 4. Hence, resultant linked list is 1->3->2->4.(REARRANGE LL)

79. Multiply 2 LL, subtraction in LL

80. Find sum of last N nodes of LL.

81. Given two string, represented as linked lists (every character is a node->data in the linked list)

. Write a function compare() that works similar to strcmp(), i.e., it returns 0 if both strings are same, 1 if first linked list is lexicographically greater, and -1 if second is lexicographically greater.(Compare 2 LL)

82. Given a linked list, the task is to move all 0s to the front of the linked list.

The order of all another element except 0 should be same after rearrangement. (MOVE ALL 0s TO FRONT OF LL)

83. LL in zigzag fashion.

84. Count triplets in LL.

85. XOR LL.

86. Insertion sort in SLL.

- 87. Partition list.
- 88. Program to insert new node in middle of circular LL.
- 89. Max and min node from circular LL
- 90. Program to swap nodes in a singly linked list without swapping data
- 91. Program to swap the last element of the singly linked list from the first one
- 92. pop operation in LL.
- 93. Given a linked list, split it into two sublists – one for the front half and one for the back half.
If the total number of elements in the list is odd, the extra element should go in the front list.