

Data Structures Open Course (Summer 2021)

Linked List problems

1. Given a singly linked list split it into two lists of equal or almost equal size. First half will be the front list and second half will be the back list.
2. Write a program to remove duplicates from a linked list sorted in increasing order.
3. Write a program to delete mth to last element of the linked list. When m=0, it means the last element of the linked list.
4. How will you sort a given unsorted singly linked list using insertion sort?
5. Given a linked list split it into two lists with alternating nodes. For example, if a list is { a, b, a, b, a } , then new lists should be {a, a, a} and { b, b}.
6. Write a function shuffle_merge which takes two linked lists and merges them taking nodes alternately from the two lists.
7. Take two lists sorted in increasing order and merge them such that the new list is also sorted.
8. Write a function sorted_intersect which takes two linked lists sorted in increasing order and returns a new list which is an intersection of the two lists.
9. Write a function that takes two lists sorted in increasing order and returns the following:
 - a. List 1 and list 2
 - b. List1 or list2
 - c. List 1 or not(list 2)
List1 and not(list 2)
10. Write a program to reverse a given Doubly Linked list.
11. Write a program to recursively reverse a singly linked list.
12. Quick sort on doubly linked list.
13. Check if linked list is palindrome recursively
14. Merge sort on singly linked list
15. Check whether a given singly linked list is cyclic or not.