Kal Academy Algorithms and Data Structures –LinkedListProblems

1.Write an algorithm to determine if a linkedlist is a palindrome

Logic: <https://youtu.be/fn3oiAc1Dug>

Code: <https://youtu.be/5ESc0LX5AgQ>

2. Write an algorithm to determine if a linkedlistis circular. FOLLOW UP: Determine where the circle meets.

Logic: <https://youtu.be/VXDdfXLOYbI>

Code: <https://youtu.be/qIoztUyPGz4>

3. Clone a linked list with a random pointer.

Logic: <https://youtu.be/OpOFdlFCAHw>

Code: Part 1: <https://youtu.be/2ilCo2Bc7ls>

Code Part 2: <https://youtu.be/kKMkqRQKR9E>

4. Write code to remove duplicates from an unsorted linked list. Follow up: How would you solve it if temporary buffer is not allowed?

Logic: <https://youtu.be/5rFemAlUt2E>

Code Part 1: <https://youtu.be/DpnK0Xe97Dc>

Coe Part 2: <https://youtu.be/JX7wgPk2SfI>

5. Implement an algorithmto find the kth to the last element of a singly linked list

Logic and Code: <https://youtu.be/zK_uPadBweU>

6. Implement an algorithm to delete a node a singly linked list, given only access to that node.

Logic and Node: <https://youtu.be/Cm4sdQrpJpQ>

7. Write code to partition a linkedlist around a value x, such that all nodes less than x come before all nodes great than or equal to x.

Logic: <https://youtu.be/HWIRv6y845U>

Code: <https://youtu.be/6l-gdd8HFaM>

8. Reverse a linked list –iteratively and recursively

Iteration : <https://youtu.be/-1A3Hkgrp3Q>

Recurrsion: <https://youtu.be/CQRfkZM3EF8>

9. Given a singly linked list which has data sorted in ascending order, construct a balanced BST.

Logic: <https://youtu.be/uQQvuVyPXIc>

Code: <https://youtu.be/k_Lpa4BQlLg>

10. Given a singly linked list, find middle of the linked list. For example, if given linked list is 1->2->3->4->5 then output should be 3. If there are even nodes, then there would be two middle nodes, we need to print second middle element. For example, if given linked list is 1->2->3->4->5->6 then output should be 4.

Logic and code <https://youtu.be/8Og01y6kER4>

11. Given a singly linked list, rotate the linked list counter-clockwise by k nodes. Where k is a given positive integer smaller than or equal to length of the linked list. For example, if the given linked list is 10->20->30->40->50->60 and k is 4, the list should be modified to 50->60->10->20->30->40.

Logic and code: <https://youtu.be/FAdF5JxfndM>

12. Given a linked list, write a function to reverse every k nodes (where k is an input to the function).If a linked list is given as 1->2->3->4->5->6->7->8->NULL and k = 3 then output will be 3->2->1->6->5->4->8->7->NULL.

Logic and code: <https://youtu.be/IlZYkDH55Xg>

13. Given two linked lists sorted in ascending order. Merge themin-place and return head of the merged list. For example, if first list is 10->20->30 and second list is 15->17, then the result list should be 10->15->17->20->30.It is strongly recommended to do merging in-place using O(1) extra space.

Logic and code Part1: <https://youtu.be/9G39E4adbQ0>

Logic and code Part 2: <https://youtu.be/XBA-efquDRw>

14. Flattening a linked list with Next and bottom pointers.

Logic: <https://youtu.be/2RPHhkmaliM>

Code: <https://youtu.be/Akv_v5bLIGo>

15. .Given a singly linked list, write a function to swap elements pairwise. For example, if the linked list is 1->2->3->4->5 then the function should change it to 2->1->4->3->5, and if the linked list is 1->2->3->4->5->6 then the function should change it to 2->1->4->3->6->5.

Logic and Code: <https://youtu.be/oHz3OWtHNAo>

16.   
Given two numbers represented by two lists, write a function that returns sum list. The sum list is list representation of addition of two input numbers.Suppose you have two linked list 5->4 ( 4 5 )and 5->4->3( 3 4 5) you have to return a resultant linked list 0->9->3 (3 9 0).

Logic: <https://youtu.be/zzpKCDkQfdY>

Code; <https://youtu.be/lzSEDTLbnaY>

‘