Assignment

(Assignment 1-19)Last date of submission + Viva : 30/01/23

(Assignment 20-30)Last date of submission + Viva : 20/02/23

1. Given an array of elements, replace every element with nearest greater element on the right of that element.

2. Recursively remove all adjacent duplicates: Given a string of characters, recursively remove adjacent duplicate characters from the string. The output string should not have any adjacent duplicates.

Input: careermonk	Input: Mississippi
Output: camonk	M

3. Given a list, List1={A_1,A_2,A_3,……..A_{n-1}, A_n} with data, reorder it to {A_1, A_n, A_2, An_{-1}, …..} without using any extra space.

4. Given a singly linked list, write a function to find root(n)th element, where n is the number of elements in the list. Assume value of n is not known in advance.

5. [**Find the 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 16^{th} node.

6. Given an array of *n* numbers, create an algorithm which displays all pairs whose sum is S.

7. If h is an hashing function and is used to hash *n* keys into a table of size *s*, where *n*<=*s*, the expected number of collisions involving a particular key X is.

8. [**Josephus circle:**] N people decided to elect a leader by arranging themselves in a circle and eliminating every m^{th} person around the circle, closing ranks as each drops out. Find which person will be the last one remaining (with rank 1).

9. For a given *k* value (*k>0*) reverse blocks of nodes in the 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.

10. Suppose there are two singly linked lists both of which intersect at some point and become a singly list. The head or starting pointers for the both 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. write a program for find a merging node.

11. Check whether the given linked list is either NULL-terminated or ends in cycle (cyclic).

12. Given an array of characters formed with a's and b's. The string is marked with special character X which represents the middle of the list (for example: ababa…ababXbabab….baaaa). Check whether the string is palindrome.

13. How do we implement two stacks using only one array? Our stack routines should not indicate an exception unless every slot in the array is used.

14. Show how to implement one queue efficiently using two stacks. Analyse the running time of queue operations.

15. Show how to implement one stack efficiently using two queues. Analyse the running time of stack operations.

16. Given a stack of integers, how do you check whether each successive pair of members in the stack is consecutive or not. The pairs can be increasing or decreasing, and if the stack has an odd number of elements, the element at the top is left out of pair. For example, if stack of elements are [4, 5, -2, -3, 11, 10, 5, 6, 20], then the output should be true because each of the pairs (4,5), (-2,-3), (11,10) and (5,6) consists of consecutive numbers.

17. Given array[] with sliding window of size *w* which is moving from the very left of the array to very right. Assume that we can only see the w numbers in the window. Each time the sliding window moves right forwards by on position. For example: The array is [1 3 -1 -3 5 3 6 7], and *w* is 3.

Window position	Max
[1 3 -1] -3 5 3 6 7	3
1 [3 -1 -3] 5 3 6 7	3
1 3 [-1 -3 5] 3 6 7	5
1 3 -1 -3 [5 3 6] 7	5
1 3 -1 -3 5 [3 6 7]	6
1 3 -1 -3 5 3 6 7	7

18. A queue is set up in a circular array A[*0,…,n-1*] with the front and rear defined as usual. Assume that *n-1* locations in the array are available for storing the elements (with the other element being used to detect full/empty condition.) Give a formula for the number of elements in the queue in terms of rear, front and n.

19. Given an integer *k* and a queue of integers, how do you reverse the order of first *k* elements of the queue, leaving the other elements in the same relative order? For example, if *k*=4 and queue have the elements [10, 20, 30, 40, 50, 60, 70, 80, 90]; the output should be [40, 30, 20, 10, 50, 60, 70, 80, 90].

20. Given a sorted doubly linked list, give an algorithm for converting it into balanced binary search tree.

21. Given a singly linked list where elements are sorted in ascending order, convert it to a height balanced binary search tree.

22. Given a BST and two numbers K1 and K2, give an algorithm for printing all elements of BST in the range of K1 and K2.

23. Construct minimal AVL trees of height 0, 1, 2, 3, 4, and 5. What is the number of nodes in a minimal AVL tree of height 6?

24. Given an AVL tree with n integer items and two integers a and b, where a and b can be any integers with a<=b. Implement an algorithm to count the number of nodes in the range [a, b].

25. Given a binary tree, find the maximum path sum. The path may start and end at any node in the tree.

26. Given a big file containing billions of numbers, how can you find the 10 maximum numbers from that file?

27. Find the shortest graph distances between every pair of vertices in a given graph. Let us assume that the given graph does not have negative edges.

28. Count the number of connected components of graph G which represented in the adjacent matrix.

29. Detecting a cycle in an undirected graph.

30. Find number of BSTs from n nodes.