

1. Implement operations (traverse, insert (beg, loc, end), delete (beg, loc, end), linear search, binary search using recursion, insertion sort) on an array.
2. Implement Sparse Array.
3. Implement insertion (at the beginning, at specified location, at the end) and deletion (at the beginning, at specified location, at the end) on single linked list and count the number of nodes & reverse the single linked list.
4. Implement insertion (at the beginning, at specified location, at the end) and deletion (at the beginning, at specified location, at the end) on circular single linked list.
5. Implement insertion (at the beginning, at specified location, at the end) and deletion (at the beginning, at specified location, at the end) on double linked list.
6. Implement insertion (at the beginning, at specified location, at the end) and deletion (at the beginning, at specified location, at the end) on circular double linked list.
7. Write a program to merge two sorted linked list and display the final sorted linked list.
8. Implement operations (push, pop) on a stack using arrays. Check the status of the stack whether there is underflow or overflow.
9. Implement the conversion of infix notation to postfix notation and evaluation of postfix notation using stacks.
10. Implement operations (enqueue, dequeue) on a queue using arrays. Check the status of the queue whether it is empty or full.
11. Implement circular queue using arrays and linked list.
12. Implement stacks and queues using linked list.
13. Implement operations on Binary Search Tree (Insertion, Deletion, Search, Traversals (using recursion)- Inorder, Preorder, Postorder).
14. Implement traversals on Binary Search Tree (using stacks) - Inorder, Preorder, Postorder).
15. Implement AVL tree.
16. Implement graph traversals (DFS & BFS)
17. Implement dijkstra's shortest path greedy algorithm and Kruskal's minimum spanning tree greedy algorithm.
18. Implement Longest Common Subsequence dynamic programming algorithm.
19. Perform various sorting techniques (heap, count, radix, bucket).
20. Perform divide and conquer sorting techniques (merge, quick).