Programming Lab – I
Advanced Data Structures
Home Assignment Program List

Submission Deadline: October 16, 2018 till midnight

1. Sometimes a program requires two stack containing the same type of items. If the two stacks are stored in separate arrays. Then one stack might overflow while there was considerable unused space in the other. A neat way to avoid the problem is to put all the space in one array and let one stack grow from one end of the array and the other stack start at the other end and grow in opposite direction i.e., toward the first stack, in this way, if one stack turns out to be large and the other small, then they will still both fit, and there will be no overflow until all the space is actually used. Declare a new structure type Double stack that includes the array and the two indices top A and top B, and write functions Push A, Push B, Pop A and Pop B to handle the two stacks with in one Double Stack.

2. Let stack_ptr be a pointer to stack of integers and item be an integer variable. Write function like Push, Pop, Initialize, Empty, and Full for doing the following tasks. [You may declare additional variable in your functions in needed]

a) Return the top element of the stack and leave the top element unchanged. If the stack is empty, return INT_MAX.
b) Return the third element from the top of the stack, provided that the stack contains at least three integers. If not, return INT_MAX. Leave the stack unchanged.
c) Returns the bottom element of stack (or INT_MAX if stack empty), and leave the stack unchanged.
d) Delete all occurrences of x from the stack, leaving the other elements of the stack in the same order.

3. Write a C program to reverse the elements in the stack using recursion.

4. Write a C program that uses stack operations to convert a given infix expression into its postfix Equivalent, Implement the stack using an array.

5. Write C++ functions to implement queues by the simple method of keeping the head of the queue always in the first position of a linear array.

6. Write a program to implement a Singly Linked List that performs the following operations:

(i) Insertion of an element in the beginning of the list
(ii) Insertion of an element in the middle of the list
(iii) Insertion of an element at the end of the list
(iv) Deletion of an element from the beginning of the list
(v) Deletion of an element from the middle of the list
(vi) Deletion of an element from the end of the list
(vii) Traverse a linked list
(viii) Finding an element in the linked list
(ix) Printing the elements of a linked list.

7. Write a C program to find the largest element in a given doubly linked list.

8. Write a program to reverse the content of a Singly Linked List.

9. Write a program to remove the duplicate element(s) in a Linked List.

10. Write a program to merge two sorted linked list.

11. Write a program to sort the contents of a singly linked list.

12. Write a C++ program that uses functions to perform the following:
a) Create a doubly linked list of integers.
b) Delete a given integer from the above doubly linked list.
c) Display the contents of the above list after deletion

13. Write C++ programs to implement a double ended queue ADT using
i) array and
ii) doubly linked list respectively.

14. Write a C++ program that uses functions to perform the following:
a) Create a binary search tree of characters.
b) Traverse the above Binary search tree recursively in Postorder.

15. Design, Develop and Implement a menu driven Program in C++ for the following operations on Binary Search Tree (BST) of Integers
a. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2
b. Traverse the BST in Inorder, Preorder and Post Order
c. Search the BST for a given element (KEY) and report the appropriate message
d. Delete an element(ELEM) from BST
e. Exit

16. Write a C++ program to perform insertion, deletion and traversal into a B-tree

17. Write a C++ program to implement all the functions of a dictionary (ADT) using hashing.

18. Write a C++ program for implementing Knuth-Morris- Pratt pattern matching algorithm.

19. Write C++ programs for implementing the following graph traversal algorithms:
a. Depth first traversal
b. Breadth first traversal

20. Write a C++ program to implement stack using linked list.

21. Write a C ++ program to count the number of nodes in the binary search tree.

22. Write a program to implement AVL Tree.

23. If we implement a queue as a circularly linked list then we need only one pointer rear (or tail) to locate both the front and rear. Write C++ functions to process a queue stored in this way.
a. Initialize
b. Add Node
c. Delete Node

24. Write a menu-driven program which will accept an array of 10 integer values and sort them with following sorting algorithms: Insertion Sort, Selection Sort, Bubble Sort, Quick sort, Merge Sort, Heap Sort, Bucket Sort, Radix Sort, and Shell Sort.

25. Write a menu-driven program which will accept an array of 10 integer values and search an elements within the array using following searching algorithms: Linear Search, Iterative Binary Search, Recursive Binary Search, Interpolation Search and Exponential Search.

26. Write a program to implement R-B tree.

27. Write a program to implement multiway tree.

28. Write a program in C++ to implement priority queue using Heap.

29. Design, develop and implement a program in C++ for the following operations on Graph(G) of Cities

a. Create a Graph of N cities using Adjacency Matrix.
b. Print all the nodes reachable from a given starting node in a digraph using BFS method
c. Check whether a given graph is connected or not using DFS method.

30. Write a program to implement Dijkstra's Algorithm priority queue.

31. Write a program to find minimum spanning tree using: Kruskal's Algorithm and Prim's Algorithm.