HERITAGE INSTITUTE OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND BUSINESS SYSTEMS

Lab Assignment

Data Structure & Algorithms Lab (CSBS 2151)
CSBS 2nd Year 1st Semester, Session: 2021 – 2022

Day 1 *(Matrix Manipulations and Sparse Matrix)*

1. Write a function to multiply two matrices. The function should check the possibility of multiplication. *Dynamic memory allocation to be used for creating matrices*.

2. Write functions to give a compressed storage representation of a sparse matrix and also find the transpose of the sparse matrix (using the compressed representation).

Additional Program:

3. Write functions to check whether a matrix is i) Lower triangular, ii) Upper triangular, iii) Diagonal, iv) Identity, v) Tridiagonal. *Dynamic memory allocation to be used for creating matrices*.

4. Write a function to reverse the elements of an array without using another array.

Day 2 (Searching)

1. Write functions to implement
   a. Linear search on an array
   b. Binary search (iterative)

Additional Program:

2. Write a function to implement Linear search with sentinel on an array.
3. Write a function to implement Interpolation Search

Day 3 (Polynomial Representation)

1. Write functions to perform the following
   a. Represent a polynomial (of single variable x)
   b. Add two polynomials
   c. Find derivative of the polynomial

Additional Program:

2. Write a function to multiply two polynomials.

Day 4 (Stack and Its Applications)

1. Write a menu driven program to implement a stack *(use an array to store the elements of the stack*).
2. Write functions to convert an infix expression to its corresponding postfix expression.

Additional Program:

3. Write a function to evaluate a postfix expression.

Day 5 (Queue and Its Types)

1. Write a menu driven program to implement a simple / linear queue (*use an array to store the elements of the queue*).
2. Write a menu driven program to implement a circular queue (*use an array to store the elements of the queue*).

   You can follow any one of the schemes A circular array with front and rear indices and one position left vacant. Or, A circular array with front and rear indices and an integer variable counting entries.

Additional Program:

3. Write a menu driven program to implement a deque using array.

   Call the two ends of the deque left and right and write four C functions, remvLeft, remvRight, insrtLeft, insrtRight to remove and insert elements at the left and right ends of the deque. Make sure that the routines work properly for empty deque and that they detect overflow and underflow

Day 6 *(Recursion)*

1. Write the following functions using recursion:
   a. GCD of two integers
   b. Linear Search
2. Write a function to implement a solution to Towers of Hanoi problem. Execute your program with number of disks = 0, 1, 2 and 3.

Additional Program:

1. Write a function using recursion to implement Binary Search
2. Write a program that recursively scrambles a word, and prints out all of the possible permutations of that word.
   For example, if the string was “123”, the permutations would be:
   “123”, “132”, “213”, “231”, “312”, and “321”

Day 7 *(Singly Linked List)*

1. Write a menu driven program to implement a singly linked list with the following operations
   a. Insert an element at any position (*front, end or intermediate*)
   b. Delete an element from any position (*front, end or intermediate*)
   c. Display the list
   d. Perform a linear search on the list.
   e. Count the number of nodes

Additional Program:

2. Add the following operations into the above program to implement a singly linked list
   f. Reverse the list, so that the last element becomes the first, and so on
   g. Concatenate two lists
3. Write a program to implement a stack using linked list, such that the operations ‘push’ and ‘pop’ are performed in constant time.
4. Write a program to implement a queue using linked list, such that the operations ‘enqueue’ and ‘dequeue’ are performed in constant time.

Day 8 *(Circular and Doubly Linked List)*

1. Write a menu driven program to implement a circular linked list with the following operations
   a. Insert an element at any position (*front, end or intermediate*)
   b. Delete an element from any position (*front, end or intermediate*)
   c. Display the list

Additional Program:

1. Write a menu driven program to implement a doubly linked list with the following operations
   a. Insert an element at any position (*front, end or intermediate*)
   b. Delete an element from any position (*front, end or intermediate*)
   c. Display the list

Day 9 *(Sorting - I)*

1. Write functions for each of the following sorting techniques (*show the result after each iteration*)
   a. Bubble
   b. Selection
   c. Insertion

Additional Program:

2. Write functions for each of the following sorting techniques
   a. Shell
   b. Radix

Day 10 *(Sorting - II)*

1. Write functions for each of the following sorting techniques
   a. Quick
   b. Merge

Additional Program:

2. Write a function to implement Heap sort (*display the list once after the initial heap is built, and then after each pass during the sorting*).

Day 11 *(Tree Operations)*

1. Write a program to construct a Binary Search Tree (BST) using a function to insert elements in BST, then perform the following
   a. Traverse the tree in preorder, inorder, postorder
   b. Search for an item in the tree

Additional Program:

1. Add the following operations into the above program of Binary Search Tree (BST)
   c. Count the number of nodes
   d. Count the number of leaves
   e. Find the height of the tree
   f. Remove a node from the tree

Day 12 *(Graph)*

1. Write a program to implement a graph using array.
2. Write two functions to implement BFS & DFS.