Lab and Home Assignment Sheet

CSEN 2111: Data Structures & Algorithms Lab

Name: ......................................................................

Roll: ......................................................................

Department of Computer Science and Engineering

Heritage Institute of Technology, Kolkata, INDIA

B.Tech 2nd Yr, Section A, Fall/Autumn - 2016

Instructors: Sandip Samaddar, Anindita kundu, Sabyasachee Banerjee

|  |  |
| --- | --- |
| CSEN 2111 Lab | 2 |

* Getting to know the rules

1. Before you step foot in the lab

Make sure that you have the assignment sheet with you.

Make sure you have the homework copy, with all assignments attempted, ready for submission.

Make sure you are not more than 15 minutes late.

Make sure that you attended 75% of the CSEN 2101 classes starting from the previous day of lab.

If any of these conditions turns out to be FALSE for you, then please stay out of the lab.

1. Prerequisites

Extensive coverage of CSEN 1201 syllabus (specially pointers, structures, dynamic memory allocation, command line argument, using rand(), srand(), clock(), etc.).

Best practices in C (indentation, naming conventions, etc.).

Basic knowledge of LINUX working environment and familiarity to GCC compiler.

Basic aptitude, interest and sincerity. NB: If you are sick or feel that you are not in the right state of the mind to attend the lab, please do not come to the lab.

If you are lagging in some of these areas, please catch up as soon as you can.

1. Rules of the lab

Please continue to sit in you allocated seat throughout the semester.

Please open LINUX environment only. ID and password will be provided on the rst day.

While leaving make sure that you have signed the attendance sheet and turned o all the devices you had turned on after entering the lab.

Try the lab assignments before coming to the lab. Don't come unprepared. Please do not copy home assignment from others.

|  |  |
| --- | --- |
| CSEN 2111 Lab | 3 |

1. Evaluation procedure

Attendance and everyday performance will add to the internal evaluation.

On the last lab day, a viva-voce will be conducted to evaluate the student's learning of the subject.

A final 60 marks final lab exam will be conducted on the dates scheduled.

For every lost assignment sheet, 4 marks will be deducted from the student's internal marks. For example, if A loses assignment sheet twice during the semester, 8 marks will be deducted.

* Introduction, Arrays, Linked Lists

Day 1: Time and Space Complexity

Lab Assignment

Create three different 10,000 x 10000 matrices matrixOne, matrixTwo and result-Matrix, using dynamic memory allocation. Initialize matrixOne and matrixTwo by using rand() or srand() function, limit the values from 0 to 9. Multiply matrixOne and matrixTwo into resultMatrix.

While execution, open another terminal and use top command to see the usage of memory by the process. Calculate the time taken for the execution of the program.

Repeat the same exercise for 100000 x 100000 matrices.

Home Assignment

Write a program (WAP) to check whether a matrix is i) identity, ii) diagonal. WAP to reverse the elements of an array without using any other variable.

Day 2: Array

Lab Assignment

WAP to add two polynomials using array. Minimize the memory usage as much as you can.

|  |  |
| --- | --- |
| CSEN 2111 Lab | 4 |

Write a program to convert a matrix into its sparse representation (triple format). Once represented in sparse format, do not revert back to the matrix format any-more. Manipulate the sparse representation to find the transpose of the matrix (which should also be in sparse representation).

Calculate and find out whether using triple format for your example is advantageous or not.

Home Assignment

WAP to multiply two polynomials using array. Minimize usage of memory.

WAP to add two matrices using sparse representation. Manipulation of data should be done in sparse format.

Day 3: Singly Linked List

Lab Assignment

Write a menu driven program to implement a singly linked list with the operations:

* 1. insert any element in any given position (front, end or intermediate)
  2. delete an element from any given position (front, end or intermediate)
  3. display the list.

Home Assignment

Write a menu driven program to implement a singly linked list with the operations:

* + 1. count the number of nodes
    2. reverse the list.
* Linear Data Structures

Day 4: Stack, Queue - with array

Lab Assignment

Write a menu driven program to implement stack, using array, with i) push, ii) pop, iii) display, iv) exit operations.

WAP to evaluate a postfix expression

Write a menu driven program to implement a queue, using array, with i) insert, ii) delete, iii) display, iv) exit operations.

|  |  |
| --- | --- |
| CSEN 2111 Lab | 5 |

Home Assignment

WAP to convert an infix expression to its corresponding post fix operation.

Write a menu driven program to implement a double-ended queue, using array, with the following operations:

* + 1. insert (from front, from rear)
    2. delete (from front, from rear)
    3. exit operations.

Day 5: Stack, Queue - with linked list

Lab Assignment

Write a menu driven program to implement a stack, using linked list, with i) push, ii) pop, iii) display, iv) exit operations.

Home Assignment

Write a menu driven program to implement a queue, using linked list, with i) insert, ii) delete, iii) display, iv) exit operations.

Day 6: Circular Queue, Deque - with linked list

Lab Assignment

i) Write a menu driven program to implement a circular linked list with the operations:

1. create the list
2. insert any element in any given position (front, end or intermediate)
3. delete an element from any given position (front, end or intermediate)
4. display the list.

ii) Write a menu driven program to implement a circular queue using linked list with i) insert, ii) delete, iii) display, iv) exit operations with overflow and underflow constraints.

Home Assignment

Write a menu driven program to implement a doubly linked list with the operations:

1. create the list
2. insert any element in any given position (front, end or intermediate)
3. delete an element from any given position (front, end or intermediate)
4. display the list.

|  |  |
| --- | --- |
| CSEN 2111 Lab | 6 |

Write a menu driven program to implement a double-ended queue, using linked list, with the following operations:

i) insert (from front, rear), ii) delete (from front, rear), iii) exit operations

* Non-linear Data Structures

Day 7: Binary Search Tree (BST)

Lab Assignment

Write a program, which creates a binary search tree (BST). Also write the functions to insert, delete (all possible cases) and search elements from a BST.

Home Assignment

Write three functions to traverse a given BST in the following orders: i) in-order, ii) pre-order, iii) post-order.

Display the elements while traversing.

* Algorithms

Day 8: Searching

Lab Assignment

WAP to implement,

i) Linear Search, ii) Binary Search (iterative), iii) Interpolation Search. Plot their running time for different size of input to compare their performance.

NB: As a pre-processing step, use bubble-sort to sort the elements in the search space.

WAP to generate integers from 1 to n (input parameter) in random order and guarantees that no number appears twice in the list. While the number sequence is being generated, store it in a text le.

Home Assignment

WAP to implement binary search recursively.

|  |  |
| --- | --- |
| CSEN 2111 Lab | 7 |

Day 9: Sorting

Lab Assignment

Write different functions for implementing,

* 1. Bubble sort,
  2. Merge sort.

Plot a graph of n vs. time taken, for n = 100, 1000, 10,000 and 100,000 to com-pare the performances of the sorting methods mentioned above. Use the second assignment of Day 9 to generate the data, using the given n values.

Home Assignment

Write different functions for implementing, i) Insertion sort, ii) selection sort iii) Quick sort.

Day 10: Sorting

Lab Assignment

Write different functions for implementing,

1. Heap sort,

Implement Priority Queue using heap.

Home Assignment

Write different functions for implementing, i) Radix sort, ii) cocktail shaker sort

Day 11: Graph Algorithms

Lab Assignment

Read a graph (consider it to be undirected) from an edge-list and store it in an adjacency list (using array).

Use the adjacency list to run DFS algorithm on the graph and print the node labels. Detect and count the number of back-edges.

Home Assignment

WAP to implement BFS algorithm of a given graph (similarly as described for DFS, instead of back-edges count cross-edges).