

Mumbai Education Trust

INSTITUTE OF ENGINEERING, NASIK
SE COMPUTER ENGINEERING DEPT.

LAB MANUAL

SEM-II

DATA STRUCTURE AND ALGORITHM
LABORATORY



Prepared by
PROF. ANAND GHARU

2023 - 24

Sr. No	Title	Page No	Date of Conduction	Date of Submission	Sign of Staff
GROUP - A					
1	A-1 Consider telephone book database of N clients. Make use of a hash table implementation to quickly look up client's telephone number. Make use of two collision handling techniques and compare them using number of comparisons required to find a set of telephone numbers				
			Don't take print of index for DSAL File		
2	A-4 To create ADT that implement the "set" concept. a. Add (new Element) -Place a value into the set , b. Remove (element) Remove the value c. Contains (element) Return true if element is in collection, d. Size () Return number of values in collection Iterator () Return an iterator used to loop over collection, e. Intersection of two sets , f. Union of two sets, g. Difference between two sets, h. Subset				
GROUP - B					
3	B-5 A book consists of chapters, chapters consist of sections and sections consist of subsections. Construct a tree and print the nodes. Find the time and space requirements of your method				
4	B-7 Construct an expression tree from the given prefix expression eg. +-a*bc/def and traverse it using post order traversal (non recursive) and then delete the entire tree.				
5	B-9 Convert given binary tree into threaded binary tree. Analyze time and space complexity of the algorithm.				
GROUP - C					
6	C-13 Represent a given graph using adjacency matrix/list to perform DFS and using adjacency list to perform BFS. Use the map of the area around the college as the graph. Identify the prominent land marks				

	as nodes and perform DFS and BFS on that.				
7	C-14 There are flight paths between cities. If there is a flight between city A and city B then there is an edge between the cities. The cost of the edge can be the time that flight take to reach city B from A, or the amount of fuel used for the journey. Represent this as a graph. The node can be represented by airport name or name of the city. Use adjacency list representation of the graph or use adjacency matrix representation of the graph. Check whether the graph is connected or not. Justify the storage representation used.				
GROUP - D					
8	D-18 Given sequence $k = k_1 < k_2 < \dots < k_n$ of n sorted keys, with a search probability p_i for each key k_i . Build the Binary search tree that has the least search cost given the access probability for each key?				
9	C-19 A Dictionary stores keywords and its meanings. Provide facility for adding new keywords, deleting keywords, updating values of any entry. Provide facility to display whole data sorted in ascending/Descending order. Also find how many maximum comparisons may require for finding any keyword. Use Height balance tree and find the complexity for finding a keyword				
GROUP - E					
10	E-22 Read the marks obtained by students of second year in an online examination of particular subject. Find out maximum and minimum marks obtained in that subject. Use heap data structure. Analyze the algorithm.				
GROUP - F					
11	F-23 Department maintains a student information. The file contains roll number, name, division and address. Allow user to add, delete information of student.				

	Display information of particular employee. If record of student does not exist an appropriate message is displayed. If it is, then the system displays the student details. Use sequential file to main the data.				
12	F-24 Company maintains employee information as employee ID, name, designation and salary. Allow user to add, delete information of employee. Display information of particular employee. If employee does not exist an appropriate message is displayed. If it is, then the system displays the employee details. Use index sequential file to maintain the data.				
MINI PROJECT/CASE STUDY					
13	Design a mini project to implement Snake and Ladders Game using Python.				

GROUP - A



GROUP - A

Practical No: 01(A-1)

Title: Consider telephone book database of N clients. Make use of a hash table implementation to quickly look up client's telephone number.

Objectives :

- To understand concept of Hashing
- To understand to find collision using hash function.
- To understand and implement collision handling techniques

Problem Statement: -

Consider telephone book database of N clients. Make use of a hash table implementation to quickly look up client's telephone number. Make use of two collision handling techniques and compare them using number of comparisons required to find a set of telephone numbers

Outcome :

- Understand & Implement hash table.
- Understand & implement collision handling techniques. .

Software Requirements :

Operating System recommended :- 64-bit Open source Linux or its derivative

Programming tools recommended :- Open Source Python, Programming tool like Jupyter Notebook, Pycharm, Spyder, G++/GCC

Hardware Requirements :

i3 or above processor , 2 GB or above RAM, 512 GB or above Hard-disk etc

Theory :

- **Definitions :**
 - **Hashing :**
 - **Hash table :**
 - **Hash function :**
 - **Collision :**
 - **Probe, Load factor, Rehashing etc.**

- **Introduction of Hashing :**
- What is Hashing?
- What is a hash Key?
- How Hashing Works?
- Advantages of Hashing
- Limitation/Disadvantages of Hashing
- Applications of Hashing.
- **Hash Function:** (with types of all hash function methods with examples)
- Hash table (with its operations and example)
- **Collision resolution (handling techniques) :** (Definition, Types, Explanation with advantages, Disadvantages and Examples for each hashing techniques with step by step solution, time complexity).

References : (don't write references in writeup, its just for references)

<https://www.javatpoint.com/hashing-in-data-structure>

<https://www.javatpoint.com/separate-chaining-for-collision-handling>

<https://www.javatpoint.com/hashing-open-addressing-for-collision-handling>

Write algorithm/pseudo code for each function.

- a) hash table creation
- b) Linear Probing
- c) Quadratic probing
- d) Double hashing

Algorithm:

Write Algorithms for program/code which you have implemented.

Flowchart :

Draw flowchart for above algorithm

Conclusion:

Thus, We have successfully implemented hash table with different collision handling techniques.

Continuous Assessment of Student :

TC	PR	IN	EC	PN	Total Marks	Faculty Signature
(2)	(2)	(2)	(2)	(2)	(10)	

– TC - Timely completion, PR - Performance, IN - Innovation, EC - Efficient Code,
PN - Punctuality and Neatness.

Practical No: 02(A-4)

Title: To create ADT that implement the "set" concept.

Objectives :

- To understand the concepts of set
- To understand ADT for Set
- To understand and implement ADT for Set operation

Problem Statement: -

To create ADT that implement the "set" concept. a. Add (new Element) -Place a value into the set , b. Remove (element) Remove the value c. Contains (element) Return true if element is in collection, d. Size () Return number of values in collection Iterator () Return an iterator used to loop over collection, e. Intersection of two sets , f. Union of two sets, g. Difference between two sets, h. Subset

Outcome :

- Understand & Implement set operation
- Understand & implement ADT . .

Software Requirements :

Operating System recommended :- 64-bit Open source Linux or its derivative

Programming tools recommended :- Open Source Python, Programming tool like Jupyter Notebook, Pycharm, Spyder, G++/GCC

Hardware Requirements :

i3 or above processor , 2 GB or above RAM, 512 GB or above Hard-disk etc

Theory :

- Concept of ADT in details.
- Explain set operation(all) with an examples

References : (don't write references in writeup, its just for references)

<https://www.geeksforgeeks.org/operations-on-sets/>

<https://www.javatpoint.com/sets-operations>

Write algorithm/pseudo code for each function

- Add (new Element) -Place a value into the set ,
- Remove (element) Remove the value
- Contains (element) Return true if element is in collection,
- Size () Return number of values in collection Iterator () Return an iterator used to loop over collection,
- Intersection of two sets ,
- Union of two sets,
- Difference between two sets,
- Subset

Algorithm:

Write Algorithms for program/code which you have implemented.

Flowchart :

Draw flowchart for above algorithm

Conclusion:

Thus, We have successfully implemented SET Concepts(ADT).

Continuous Assessment of Student :

TC	PR	IN	EC	PN	Total Marks	Faculty Signature
(2)	(2)	(2)	(2)	(2)	(10)	

- TC - Timely completion, PR - Performance, IN - Innovation, EC - Efficient Code,
PN - Punctuality and Neatness.

GROUP - B



GROUP - B

Practical No: 03(B-5)

Title: Write C++ program to Print the tree level wise.

Objectives :

- To understand concept of tree data structure
- To understand concept & features of object oriented programming.

Problem Statement: -

A book consists of chapters, chapters consist of sections and sections consist of subsections. Construct a tree and print the nodes. Find the time and space requirements of your method.

Outcome :

- Define class for structures using Object Oriented features.
- Analyze tree data structure..

Software Requirements :

Operating System recommended :- 64-bit Open source Linux or its derivative

Programming tools recommended :- Open Source G++/GCC

Hardware Requirements :

i3 or above processor , 2 GB or above RAM, 512 GB or above Hard-disk etc

Theory :

- State all terminologies of tree
- Concept of tree in details (with advantages, Limitation, applications and example)

References : (don't write references in writeup, its just for references)

<https://www.geeksforgeeks.org/introduction-to-tree-data-structure-and-algorithm-tutorials/>

<https://www.javatpoint.com/tree>

Write algorithm/pseudo code for each function

1. Create tree
2. Display tree

Algorithm:

Write Algorithms for program/code which you have implemented.

Flowchart :

Draw flowchart for above algorithm

Conclusion:

Thus, We have successfully implement tree data structure.

Continuous Assessment of Student :

TC	PR	IN	EC	PN	Total Marks	Faculty Signature
(2)	(2)	(2)	(2)	(2)	(10)	

- TC - Timely completion, PR - Performance, IN - Innovation, EC - Efficient Code,
PN - Punctuality and Neatness.



Practical No: 04(B-6)

Title: Write C++ program to Construct Binary Search Tree.

Objectives :

- To understand concept of tree data structure
- To understand concept & features of object oriented programming.

Problem Statement: -

Beginning with an empty binary search tree, Construct binary search tree by inserting the values in the order given. After constructing a binary tree - i. Insert new node, ii. Find number of nodes in longest path from root, iii. Minimum data value found in the tree, iv. Change a tree so that the roles of the left and right pointers are swapped at every node, v. Search a value.

Outcome :

- Binary search tree created.
- Analyze tree data structure.
- Performed operation on tree

Software Requirements :

Operating System recommended :- 64-bit Open source Linux or its derivative

Programming tools recommended :- Open Source G++/GCC

Hardware Requirements :

i3 or above processor , 2 GB or above RAM, 512 GB or above Hard-disk etc

Theory :

- **Introduction of Binary Search Tree : (Definition, Concepts(Working) or algorithms, Operations of BST, advantages, Limitation, Applications and example).**
- **Time Complexity of Binary Search Tree**
- **Space Complexity of Binary Search Tree**

References : (don't write references in writeup, its just for references)

<https://www.javatpoint.com/binary-search-tree>

<https://www.programiz.com/dsa/binary-search-tree>

Write algorithm/pseudo code for each function

- i. Insert new node,
- ii. Find number of nodes in longest path from root,
- iii. Minimum data value found in the tree,
- iv. Change a tree so that the roles of the left and right pointers are swapped at every node,
- v. Search a value.

Algorithm:

Write Algorithms for program/code which you have implemented.

Flowchart :

Draw flowchart for above algorithm

Conclusion:

Thus, We have successfully implemented Binary Search Tree and Performed Various Operation on tree.

Continuous Assessment of Student :

TC	PR	IN	EC	PN	Total Marks	Faculty Signature
(2)	(2)	(2)	(2)	(2)	(10)	

- TC - Timely completion, PR - Performance, IN - Innovation, EC - Efficient Code,
PN - Punctuality and Neatness.

Practical No: 05(B-7)

Title: Write C++ program to Construct Expression Tree from prefix expression.

Objectives :

1. To Understand the concept of expression tree and binary tree.
2. To Understand the different type of traversals (recursive & non-recursive).
3. To understand the different notations like prefix, infix and postfix and the conversion between them using data structures (like stack and tree).

Problem Statement: -

Construct an expression tree from the given prefix expression eg. $+-a*bc/def$ and traverse it using post order traversal (non recursive) and then delete the entire tree.

Outcome :

- Define class for structures using Object Oriented features.
- Analyze stack / tree data structure.
- Postorder expression for given prefix expression

Software Requirements :

Operating System recommended :- 64-bit Open source Linux or its derivative

Programming tools recommended :- Open Source G++/GCC

Hardware Requirements :

i3 or above processor , 2 GB or above RAM, 512 GB or above Hard-disk etc

Theory :

- **Expression : (Definition, Concepts, Types of Expression(infix, Prefix and postfix with examples)**
- **Stack Data structure approach :(concepts, operations, ADT of stack)**
- **Tree data structure approach : (Explain different traversal techniques with algorithms and examples)**
- **Compare Stack Vs Tree**

References : (don't write references in writeup, its just for references)

<https://www.javatpoint.com/expression-tree-in-data-structure>

<https://www.simplilearn.com/tutorials/data-structure-tutorial/stacks-in-data-structures>

<https://www.javatpoint.com/tree-traversal>

<https://runestone.academy/ns/books/published/pythonds/BasicDS/InfixPrefixandPostfixExpressions.html>

Write algorithm/pseudo code for each function

1. Prefix expression
2. Non-recursive postorder
3. Delete tree
4. Display

Algorithm:

Write Algorithms for program/code which you have implemented.

Flowchart :

Draw flowchart for above algorithm

Conclusion:

Thus, We have successfully implemented Expression tree and obtained output as postfix expression.

Continuous Assessment of Student :

TC	PR	IN	EC	PN	Total Marks	Faculty Signature
(2)	(2)	(2)	(2)	(2)	(10)	

- TC - Timely completion, PR - Performance, IN - Innovation, EC - Efficient Code,
PN - Punctuality and Neatness.

GROUP - C



GROUP - C

Practical No: 06(C-13)

Title: Write C++ program to Represent a given graph using adjacency matrix to perform DFS and BFS.

Objectives :

1. To Understand the concept of graph data structure.
2. To Understand the different type of traversals (recursive & non-recursive).
3. To understand the DFS and BFS traversal

Problem Statement: -

Represent a given graph using adjacency matrix/list to perform DFS and using adjacency list to perform BFS. Use the map of the area around the college as the graph. Identify the prominent land marks as nodes and perform DFS and BFS on that.

Outcome :

- Define class for structures using Object Oriented features.
- Analyze DFS and BFS Traversal.
- Display Adjacency matrix, DFS and DFS traversal path.

Software Requirements :

Operating System recommended :- 64-bit Open source Linux or its derivative

Programming tools recommended :- Open Source G++/GCC

Hardware Requirements :

i3 or above processor , 2 GB or above RAM, 512 GB or above Hard-disk etc

Theory :

- **Graph Data Structure: (Definition, Concepts, Types of graph, Operations, example and applications)**
- **Representation of Adjacency Matrix and List with example**
- **DFS and BFS Traversal techniques with advantages, disadvantages, examples and algorithms and applications**
- **Compare DFS and BFS**

References : (don't write references in writeup, its just for references)

<https://www.javatpoint.com/ds-graph>

<https://www.javatpoint.com/bfs-vs-dfs>

[https://unacademy.com/content/gate-cse-it/difference-between-bfs-and-dfs/#:~:text=BFS%20\(Breadth%20First%20Search\)%20finds,of%20the%20Stack%20data%20s,tructure.&text=Because%20BFS%20reaches%20a%20vertex,path%20inside%20an%20unweigh,ted%20graph.](https://unacademy.com/content/gate-cse-it/difference-between-bfs-and-dfs/#:~:text=BFS%20(Breadth%20First%20Search)%20finds,of%20the%20Stack%20data%20s,tructure.&text=Because%20BFS%20reaches%20a%20vertex,path%20inside%20an%20unweigh,ted%20graph.)

Write algorithm/pseudo code for each function

1. Accept graph
2. Display adjacency matrix
3. Display DFS Traversal
4. Display BFS Traversal
- 5.

Algorithm:

Write Algorithms for program/code which you have implemented.

Flowchart :

Draw flowchart for above algorithm

Conclusion:

Thus, We have successfully implemented Graph using adjacency matrix.

Continuous Assessment of Student :

TC	PR	IN	EC	PN	Total Marks	Faculty Signature
(2)	(2)	(2)	(2)	(2)	(10)	

- TC - Timely completion, PR - Performance, IN - Innovation, EC - Efficient Code,
PN - Punctuality and Neatness.

Practical No: 07(C-14)

Title: Write C++ program to Represent a given graph using adjacency matrix.

Objectives :

1. To understand concept of Graph data structure
2. Understand & implement adjacency matrix for graph

Problem Statement: -

There are flight paths between cities. If there is a flight between city A and city B then there is an edge between the cities. The cost of the edge can be the time that flight take to reach city B from A, or the amount of fuel used for the journey. Represent this as a graph. The node can be represented by airport name or name of the city. Use adjacency list representation of the graph or use adjacency matrix representation of the graph. Justify the storage representation used..

Outcome :

- Learn concepts of Graph Data Structure
- Analyze working of functions
- Display Adjacency matrix, DFS and DFS traversal path.

Software Requirements :

Operating System recommended :- 64-bit Open source Linux or its derivative

Programming tools recommended :- Open Source G++/GCC

Hardware Requirements :

i3 or above processor , 2 GB or above RAM, 512 GB or above Hard-disk etc

Theory :

- **Graph Data Structure: (Definition, Concepts, Draw graph for given problem statement i.e. node as cities and weight as time etc)**
- **Representation of Adjacency Matrix with example(for above graph)**
- **Undirected graph definition**
- **Compare DFS and BFS**

References : (don't write references in writeup, its just for references)

<https://www.javatpoint.com/ds-graph>

<https://www.javatpoint.com/bfs-vs-dfs>

[https://unacademy.com/content/gate-cse-it/difference-between-bfs-and-dfs/#:~:text=BFS%20\(Breadth%20First%20Search\)%20finds,of%20the%20Stack%20data%20s tructure.&text=Because%20BFS%20reaches%20a%20vertex,path%20inside%20an%20unweigh](https://unacademy.com/content/gate-cse-it/difference-between-bfs-and-dfs/#:~:text=BFS%20(Breadth%20First%20Search)%20finds,of%20the%20Stack%20data%20s tructure.&text=Because%20BFS%20reaches%20a%20vertex,path%20inside%20an%20unweigh)

Write algorithm/pseudo code for each function

1. Accept graph
2. Display adjacency matrix
3. Display DFS Traversal
4. Display BFS Traversal
- 5.

Algorithm:

Write Algorithms for program/code which you have implemented.

Flowchart :

Draw flowchart for above algorithm

Conclusion:

Thus, This program gives us the knowledge of adjacency matrix graph.

Continuous Assessment of Student :

TC	PR	IN	EC	PN	Total Marks	Faculty Signature
(2)	(2)	(2)	(2)	(2)	(10)	

- TC - Timely completion, PR - Performance, IN - Innovation, EC - Efficient Code, PN - Punctuality and Neatness.

GROUP - D



GROUP - D

Practical No: 08(C-18)

Title: Write C++ program to implement OBST Tree.

Objectives :

1. To understand concept of OBST.
2. To understand concept & features like extended binary search tree.

Problem Statement: -

Given sequence $k = k_1 < k_2 < \dots < k_n$ of n sorted keys, with a search probability p_i for each key k_i . Build the Binary search tree that has the least search cost given the access probability for each key?.

Outcome :

- Define class for Extended binary search tree using Object Oriented features.
- Analyze working of functions..

Software Requirements :

Operating System recommended :- 64-bit Open source Linux or its derivative

Programming tools recommended :- Open Source G++/GCC

Hardware Requirements :

i3 or above processor , 2 GB or above RAM, 512 GB or above Hard-disk etc

Theory :

- Dynamic Programming
- Principle of Optimality
- OBST: (Definition, Concepts, advantages, disadvantages, applications and example)
- Algorithms of OBST

References : (don't write references in writeup, its just for references)

<https://www.javatpoint.com/optimal-binary-search-tree>

<https://www.geeksforgeeks.org/optimal-binary-search-tree-dp-24/>

<https://www.tutorialspoint.com/optimal-binary-search-trees-in-data-structures>

Write algorithm/pseudo code for each function

Write algo for your code.(if available)

Algorithm:

Write Algorithms for program/code which you have implemented.

Flowchart :

Draw flowchart for above algorithm

Conclusion:

Thus, we have successfully implemented OBST tree

Continuous Assessment of Student :

TC	PR	IN	EC	PN	Total Marks	Faculty Signature
(2)	(2)	(2)	(2)	(2)	(10)	

- TC - Timely completion, PR - Performance, IN - Innovation, EC - Efficient Code,
PN - Punctuality and Neatness.



Practical No: 09(C-19)

Title: Write C++ program to implement Height Balance Tree.

Objectives :

1. To understand concept of height balanced tree data structure.
2. To understand procedure to create height balanced tree.

Problem Statement: -

A Dictionary stores keywords and its meanings. Provide facility for adding new keywords, deleting keywords, updating values of any entry. Provide facility to display whole data sorted in ascending/Descending order. Also find how many maximum comparisons may require for finding any keyword. Use Height balance tree and find the complexity for finding a keyword

Outcome :

- Define class for AVL using Object Oriented features.
- Analyze working of various operations on AVL Tree

Software Requirements :

Operating System recommended :- 64-bit Open source Linux or its derivative

Programming tools recommended :- Open Source G++/GCC

Hardware Requirements :

i3 or above processor , 2 GB or above RAM, 512 GB or above Hard-disk etc

Theory :

- **AVL Tree : (Definition, Concepts, advantages, disadvantages, applications)**
- **Types of AVL Rotation with example**
- **Algorithms of AVL Tree, example of AVL Tree**

References : (don't write references in writeup, its just for references)

<https://www.geeksforgeeks.org/introduction-to-avl-tree/>

<https://www.javatpoint.com/avl-tree>

Write algorithm/pseudo code for each function

1. Add new keyword
2. Delete keyword
3. Update keyword
4. Display Result(Ascending/Descending)
5. Maximum number of comparison

Algorithm:

Write Algorithms for program/code which you have implemented.

Flowchart :

Draw flowchart for above algorithm

Conclusion:

Thus, This program gives us the knowledge height balanced binary tree.

Continuous Assessment of Student :

TC	PR	IN	EC	PN	Total Marks	Faculty Signature
(2)	(2)	(2)	(2)	(2)	(10)	

- TC - Timely completion, PR - Performance, IN - Innovation, EC - Efficient Code,
PN - Punctuality and Neatness.

GROUP - E



GROUP - E

Practical No: 07(C-14)

Title: Write C++ program to implement Heap Data Structure

Objectives :

1. To understand concept of Heap data structure
2. Understand & implement Minheap and Maxheap function and perform heap sort

Problem Statement: -

Read the marks obtained by students of second year in an online examination of particular subject. Find out maximum and minimum marks obtained in that subject. Use heap data structure. Analyze the algorithm.

Outcome :

- Define class for heap Data Structure
- Perform heap sort
- Display Minheap and Maxheap.

Software Requirements :

Operating System recommended :- 64-bit Open source Linux or its derivative

Programming tools recommended :- Open Source G++/GCC

Hardware Requirements :

i3 or above processor , 2 GB or above RAM, 512 GB or above Hard-disk etc

Theory :

- **Heap Data Structure: (Definition, Concepts, representation, advantages, disadvantages, applications and example)**
- **State Heapify, Minheap and Maxheap with representation and example.**
- **Write algorithm for heap implementation(Maxheap and Minheap)**

References : (don't write references in writeup, its just for references)

<https://www.geeksforgeeks.org/heap-data-structure/>

<https://www.javatpoint.com/heap-data-structure>

Write algorithm/pseudo code for each function

1. Accept marks of student
2. Perform heapify function
3. Perform HeapSort function
4. Display MinHeap and MaxHeap

Algorithm:

Write Algorithms for program/code which you have implemented.

Flowchart :

Draw flowchart for above algorithm

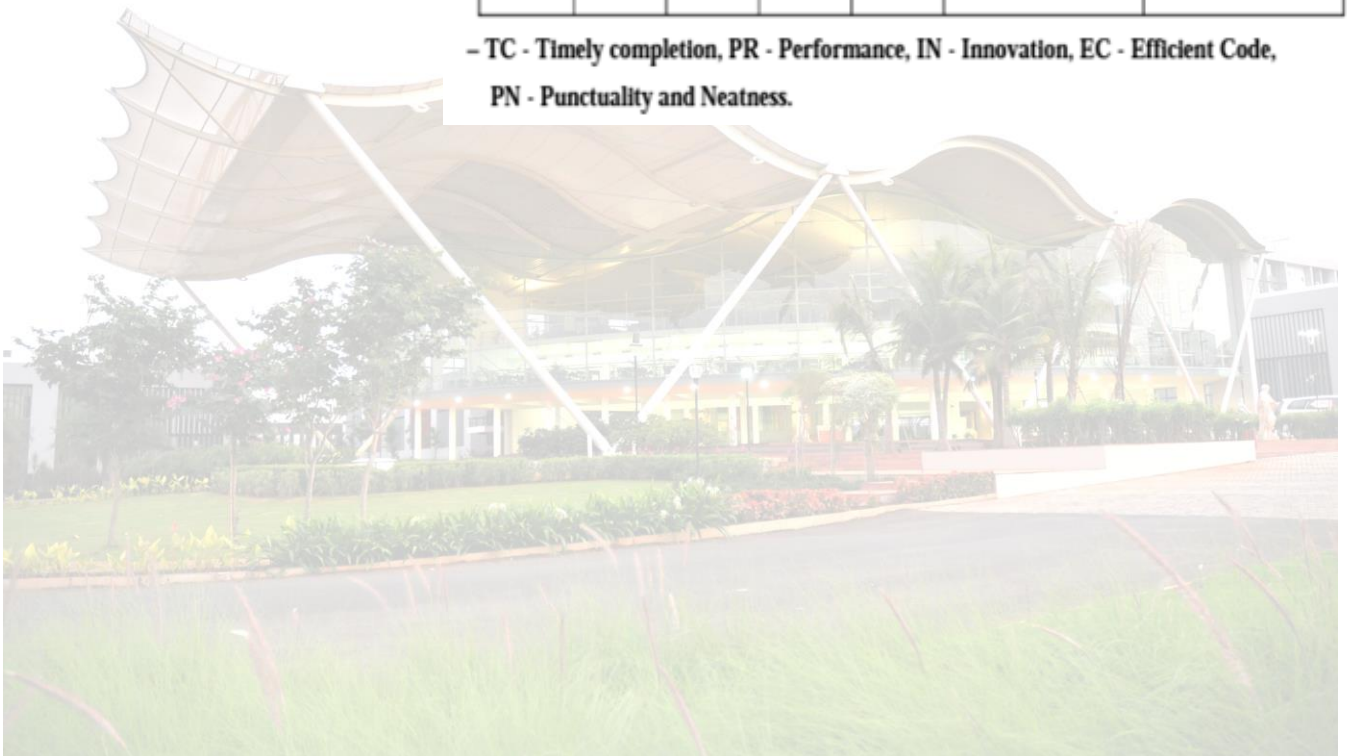
Conclusion:

Thus, we have successfully implemented Heap data structure with Minheap and Maxheap..

Continuous Assessment of Student :

TC	PR	IN	EC	PN	Total Marks	Faculty Signature
(2)	(2)	(2)	(2)	(2)	(10)	

- TC - Timely completion, PR - Performance, IN - Innovation, EC - Efficient Code,
PN - Punctuality and Neatness.



GROUP - F



GROUP - F

Practical No: 11(F-23)

Title: To implement the concept of sequential file and to perform basic operation as adding record, display all record, search record from sequential file.

Objectives :

- To understand concept of file organization in data structure.
- To understand concept & features of sequential file organization.

Problem Statement: -

Department maintains a student information. The file contains roll number, name, division and address. Allow user to add, delete information of student. Display information of particular employee. If record of student does not exist an appropriate message is displayed. If it is, then the system displays the student details. Use sequential file to main the data.

Outcome :

- Define class for sequential file using Object Oriented features.
- Analyze working of various operations on sequential file.

Software Requirements :

Operating System recommended :- 64-bit Open source Linux or its derivative

Programming tools recommended :- Open Source G++/GCC

Hardware Requirements :

i3 or above processor , 2 GB or above RAM, 512 GB or above Hard-disk etc

Theory :

- **File Organization :**
- **Types/Methods of File Organization :**
- **Explain Sequential File Organization (Definition, Concepts, representation, Operations, advantages, disadvantages, applications and example)**

The I/O system of C++ contains a set of classes that define the file handling methods.

These include ifstream, ofstream and fstream . These classes are derived from fstream base and from corresponding stream classes. These classes are declared in fstream.h header file.

We must include this file in the program that uses file.(Explain in details from sent pdf)

References : (don't write references in writeup, its just for references)

<https://www.geeksforgeeks.org/file-organization-in-dbms-set-1/>

<https://www.geeksforgeeks.org/sequential-file-organization-in-dbms/>

Write algorithm/pseudo code for each function

1. Accept student information
2. Add information of student
3. Delete information of student
4. Display information of student

Algorithm:

Write Algorithms for program/code which you have implemented.

Flowchart :

Draw flowchart for above algorithm

Conclusion:

Thus, we have successfully implemented operations of sequential file organization.

Continuous Assessment of Student :

TC	PR	IN	EC	PN	Total Marks	Faculty Signature
(2)	(2)	(2)	(2)	(2)	(10)	

- TC - Timely completion, PR - Performance, IN - Innovation, EC - Efficient Code,
PN - Punctuality and Neatness.

Practical No: 12(F-24)

Title: To implement the concept of Index sequential file and to perform basic operation as adding record, display all record, search record.

Objectives :

- To understand concept of file organization in data structure.
- To understand concept & features of Indexed Sequential file organization.

Problem Statement: -

Company maintains employee information as employee ID, name, designation, and salary. Allow user to add, delete information of employee. Display information of particular employee. If employee does not exist an appropriate message is displayed. If it is, then the system displays the employee details. Use index sequential file to maintain the data.

Outcome :

- Define class for sequential file using Object Oriented features.
- Analyze working of various operations on Indexed sequential access method

Software Requirements :

Operating System recommended :- 64-bit Open source Linux or its derivative

Programming tools recommended :- Open Source G++/GCC

Hardware Requirements :

i3 or above processor , 2 GB or above RAM, 512 GB or above Hard-disk etc

Theory :

- **Indexed Sequential Access File Organization : (Definition, Concepts, representation, Operations, advantages, disadvantages, applications and example)**
- **Difference between Sequential vs Indexed Sequential File Organization**

References : (don't write references in writeup, its just for references)

<https://www.javatpoint.com/dbms-indexed-sequential-access-method>

<https://www.geeksforgeeks.org/difference-between-sequential-indexed-and-relative-files-in-cobol/>

Write algorithm/pseudo code for each function

1. Accept Employee information
2. Add information of Employee
3. Delete information of Employee
4. Display information of Employee

Algorithm:

Write Algorithms for program/code which you have implemented.

Flowchart :

Draw flowchart for above algorithm

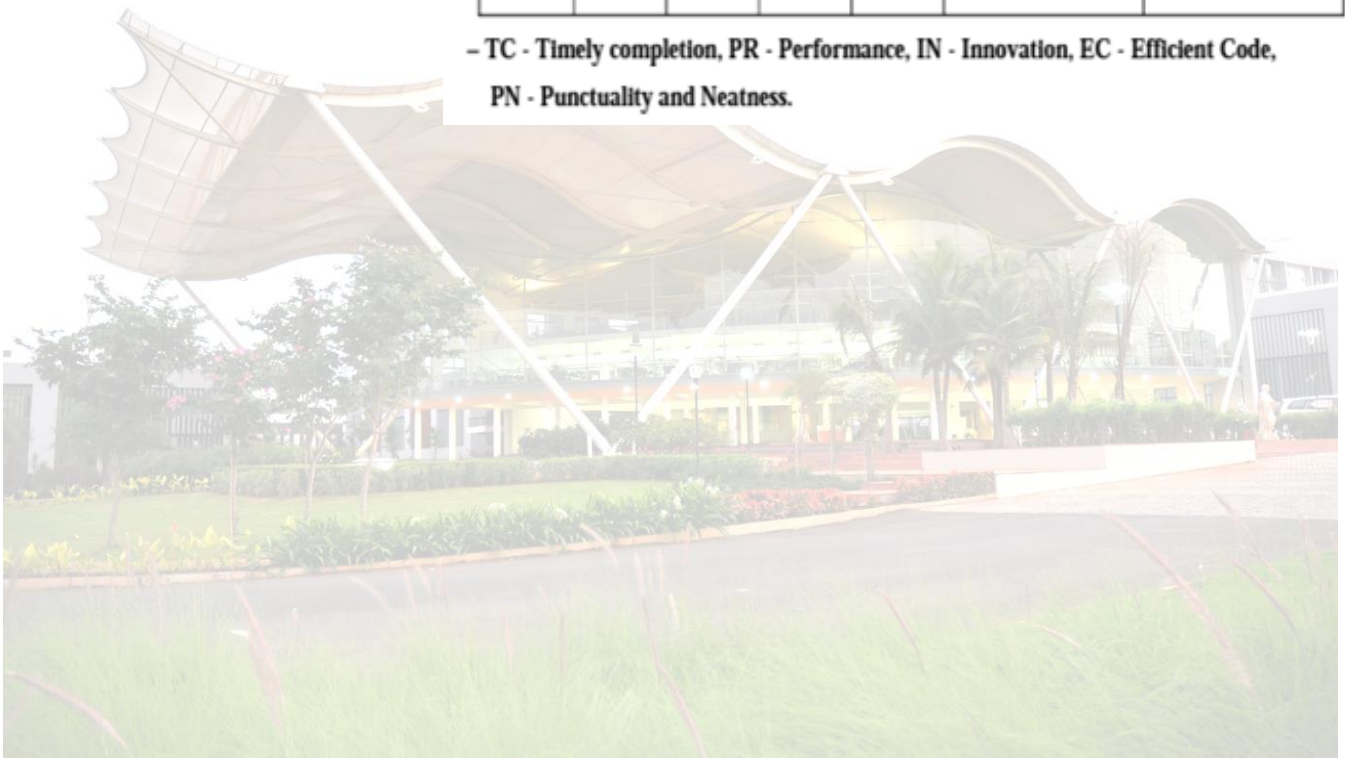
Conclusion:

Thus, we have successfully implemented operations of Indexed Sequential file Organization.

Continuous Assessment of Student :

TC	PR	IN	EC	PN	Total Marks	Faculty Signature
(2)	(2)	(2)	(2)	(2)	(10)	

- TC - Timely completion, PR - Performance, IN - Innovation, EC - Efficient Code,
PN - Punctuality and Neatness.



Mini Project



Mini Project

Practical No: 13(G-28)

Title: To implement Snake and Ladders Game using Python.

Objectives :

- To understand concept & features of Python programming.

Problem Statement: -

Design a mini project to implement Snake and Ladders Game using Python.

Outcome :

- Implementation of Snake and Ladders Game using Python.

Software Requirements :

Operating System recommended :- 64-bit Open source Linux or its derivative

Programming tools recommended :- Open Source python, Pycharm IDE

Hardware Requirements :

i3 or above processor , 2 GB or above RAM, 512 GB or above Hard-disk etc

Theory :

Python:

- Python is a high-level, general-purpose and a very popular programming language.
- Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting-edge technology in Software Industry.
- Python Programming Language is very well suited for Beginners, also for experienced programmers with other programming languages like C++ and Java.

Python Programming Language:

1. Python is currently the most widely used multi-purpose, high-level programming language.
2. Python allows programming in Object-Oriented and Procedural paradigms.
3. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.
4. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber... etc.
5. The biggest strength of Python is huge collection of standard library which can be used for the following:

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia
- Scientific computing

This Snakes and Ladders Game contains python file scripts (snakes_ladders.py), resources files and sound files. The gameplay of the system, which is the user can choose an option either to play multiple participant or with the computer.

Beginning of the game, the player needs to roll the dice and in the wake of moving it the game moves the token consequently as indicated by the dice number. The interactivity is like the genuine one. Here, the player likewise gets one more opportunity to roll the dice at whatever point he/she gets 6 number.

There are quantities of stepping stools and snakes in the game which causes the player to update or minimization the square number. The player who arrives at the last square of the track is the champ.

Designing a complete Snake and Ladder game using object-oriented programming (Python OOP) principles with the following rules and requirements :

Rules for game :

1. Here we create a board of size 10 and dice of side 6.
2. Each player puts their counter on the board at starting position at 1 and takes turns to roll the dice.
3. Move your counter forward the number of spaces shown on the dice.
4. If your counter lands at the bottom of a ladder, you can move up to the top of the ladder. If your counter lands on the head of a snake, you must slide down to the bottom of the snake.
5. Each player will get a fair chance to roll the dice.
6. On the dice result of 6, the user gets one more chance to roll the dice again. However, the same user can throw the dice a maximum of 3 times.

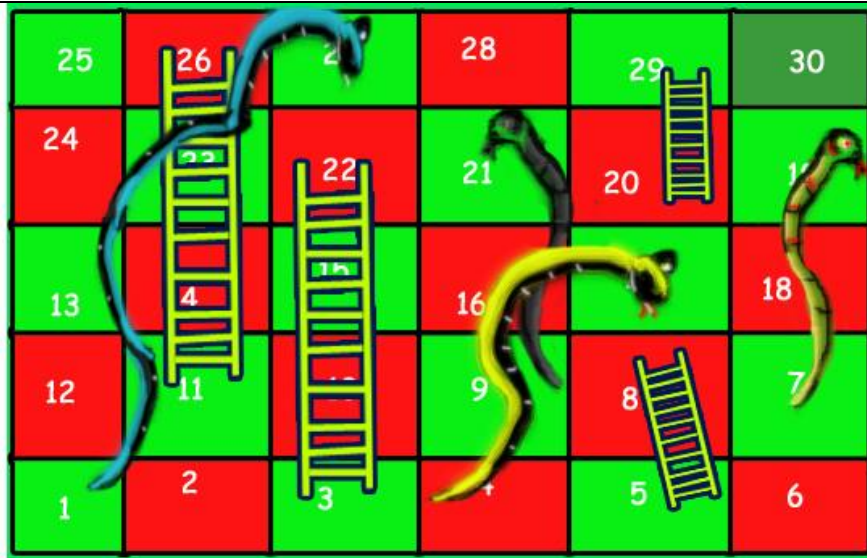
Note: if the result of the dice is 6,6,6 the user can not throw the dice again as the maximum attempts are over and the next user will get to throw the dice.

7. When the user rolls dice and it leads to an invalid move, the player should remain in the same position.

Ex: when the user is in position 99 and rolling of dice yields any number more than one the user remains in the same position.

8. Print the ranks of users who finished first, second, and so on...

Note : Take print of image and stick to writeup



Step to design Snakes and Ladders Game

- Step 1: Create a project name.

First when you finished installed the Pycharm IDE in your computer, open it and then create a “project name” after creating a project name click the “create” button.

- Step 2: Create a python file.

Second after creating a project name, “right click” your project name and then click “new” after that click the “python file“.

- Step 3: Name your python file.

Third after creating a python file, Name your python file after that click “enter“.

- Step 4: The Python code.

The actual coding of how to create Snakes and Ladders Game in Python

References : (don't write references in writeup, its just for references)

<https://www.geeksforgeeks.org/design-snake-and-ladder-game-using-python-oops/>

<https://gist.github.com/theoctober19th/6f7d93a59dcc644a4345900cc18c99dd>

Algorithm:

Write Algorithms for program/code which you have implemented.

Flowchart :

Draw flowchart for above algorithm

Conclusion:

Thus, we have successfully implemented operations of Indexed Sequential file Organization.

Continuous Assessment of Student :

TC	PR	IN	EC	PN	Total Marks	Faculty Signature
(2)	(2)	(2)	(2)	(2)	(10)	

– TC - Timely completion, PR - Performance, IN - Innovation, EC - Efficient Code, PN - Punctuality and Neatness.



THANKS...!