SIX WEEKS SUMMER TRAINING REPORT

ON

C++ AND DATA STRUCTURES ALGORITHMS

SUBMITTED BY:

Name: Tanmay Anand

Registration No: 12114719

Programme: B.Tech CSE

Under the Guidance of

Cipher School & Anurag Sir

School of Computer Science & Engineering

Lovely Professional University, Phagwara

Period of June-Aug'23.

DECLARATION

I hereby declare that I have completed my six weeks summer training at **Cipher Schools online Course** and **VIP Road, CCC, Office Number: 670, Zirakpur - 140603** from **June 2023** to **Aug 2023** under the guidance of **Anurag Sir.** I have declare that I have worked with full dedication during these six weeks of training and my learning outcomes fulfill the requirements of training for the award of degree of **BTech CSE**, Lovely Professional University, Phagwara.

Date : 10/07/2023

Tanmay Anand

(12114719)

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my **CipherSchools Industry Coordinator(Anurag Sir)**who gave me a golden opportunity to do this project and also provided support in completing in my project.

I would also like to extend my gratitude to our **Lovely Professional University** for providing me with all the facilities that were required.

Tanmay Anand

(12114719)

CERTIFICATE

TABLE OF CONTENTS

ABBREVIATION

The abbreviation of **"DSA"** is commonly used to refer to **"Data Structures and Algorithms,"** and the abbreviation of **"C++"** stands for the programming language itself.

INTRODUCTION

The course name **DSA** stands for **"Data Structures and Algorithms**" and Self- paced means, one can join the course anytime. All of the content will be available once one gets enrolled. One can finish it at his own decided speed.

What is Data Structure?

Data Structure is a way of collecting and organizing data in such a way that we can perform operations on these data in an effective way. Data Structures is about rendering data elements in terms of some relationship, for better organization and storage. For example, we have some data which has, player's **name** "Virat" and **age** 26. Here "Virat" is of **String** data type and 26 is of **integer** data type.

What is Algorithm?

An algorithm is a finite set of instructions or logic, written in order, to accomplish a certain predefined task. Algorithm is not the complete code or program, it is just the core logic(solution) of a problem, which can be expressed either as an informal high-level description as **pseudocode** or using a **flowchart**.

Importance of DSA:

Understanding data structures and algorithms is fundamental for becoming a proficient programmer and problem solver. Here's why DSA is important:

Efficiency: Knowing different data structures and algorithms allows you to choose the most appropriate one for a specific problem, which can significantly impact the efficiency of your program in terms of time and memory usage.

Problem Solving: DSA provides a systematic approach to solving problems. By breaking down complex problems into smaller, manageable steps, you can devise solutions more effectively.

Coding Interviews: Many technical interviews for software engineering positions involve questions related to DSA. Familiarity with these concepts can greatly enhance your performance in coding interviews.

Optimization: Efficient algorithms are crucial for optimizing processes, whether in software applications, data analysis, or other computational tasks.

Software Design: Designing effective software often requires choosing appropriate data structures and algorithms to ensure the application's performance and scalability.

Critical Thinking: DSA encourages you to think critically about problem-solving approaches and trade-offs. It helps you develop a structured mindset for tackling complex challenges.

This course is a complete package that helped me learn Data Structures and Algorithms from basic to an advanced level. The course curriculum has been divided into 8 weeks where one can practice questions & attempt the assessment tests according to his own pace. The course offers me a wealth of programming challenges that will help me to prepare for interviews with top-notch companies like Microsoft, Amazon, Adobe etc.

TECHNOLOGY LEARNT

It had 24 units which was further divided into chapters and then topics so during my whole 8 week course I learned the following :

INTRODUCTION TO DSA

• Analysis of Algorithm

o In this I learned about background analysis through a Program and its functions.

• Order of Growth

o A mathematical explanation of the growth analysis through limits and functions.

o A direct way of calculating the order of growth

• Asymptotic Notations

o Best, Average and Worst case explanation through a program.

• Big O Notation

o Graphical and mathematical explanation.

o Calculation o Applications at Linear Search

• Omega Notation

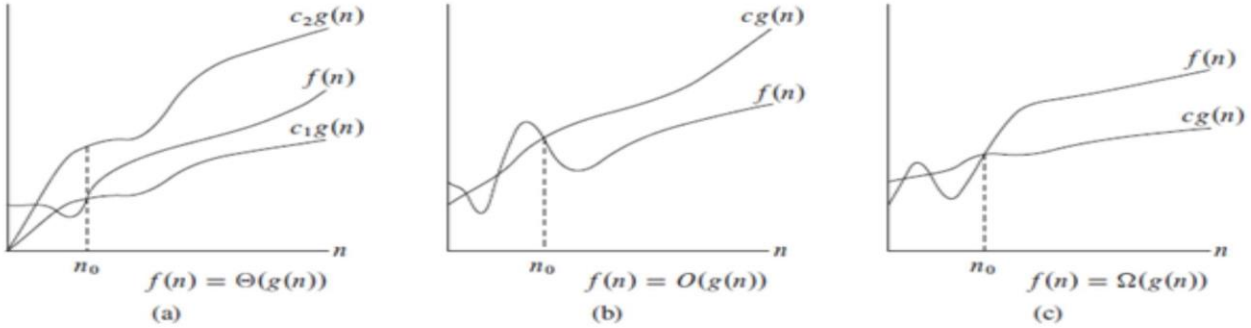
o Graphical and mathematical explanation.

o Calculation.

• Theta Notation

o Graphical and mathematical explanation.

o Calculation.



• Analysis of common loops

o Single, multiple and nested loops

• Analysis of Recursion

o Various calculations through Recursion Tree method

• Space Complexity

o Basic Programs

o Auxiliary Space

o Space Analysis of Recursion

o Space Analysis of Fibonacci number

MATHEMATICS

- **Finding the number of digits in a number.**
- **Arithmetic and Geometric Progressions.**
- **Quadratic Equations.**
- **Mean and Median.**
- **Prime Numbers.**
- **LCM and HCF**
- **Factorials**
- **Permutations and Combinations**
- **Modular Arithmetic**

BITMAGIC

- **Bitwise Operators in C++**

o Operation of AND, OR, XOR operators

o Operation of Left Shift, Right Shift and Bitwise Not

- **Bitwise Operators in Java**

o Operation of AND, OR

o Operation of Bitwise Not, Left Shift

- Operation of Right Shift and unsigned Right Shift

- **Problem(With Video Solutions): Check Kth bit is set or not**

- Method 1: Using the left Shift.

- Method 2: Using the right shift

RECURSION

- **Introduction to Recursion**

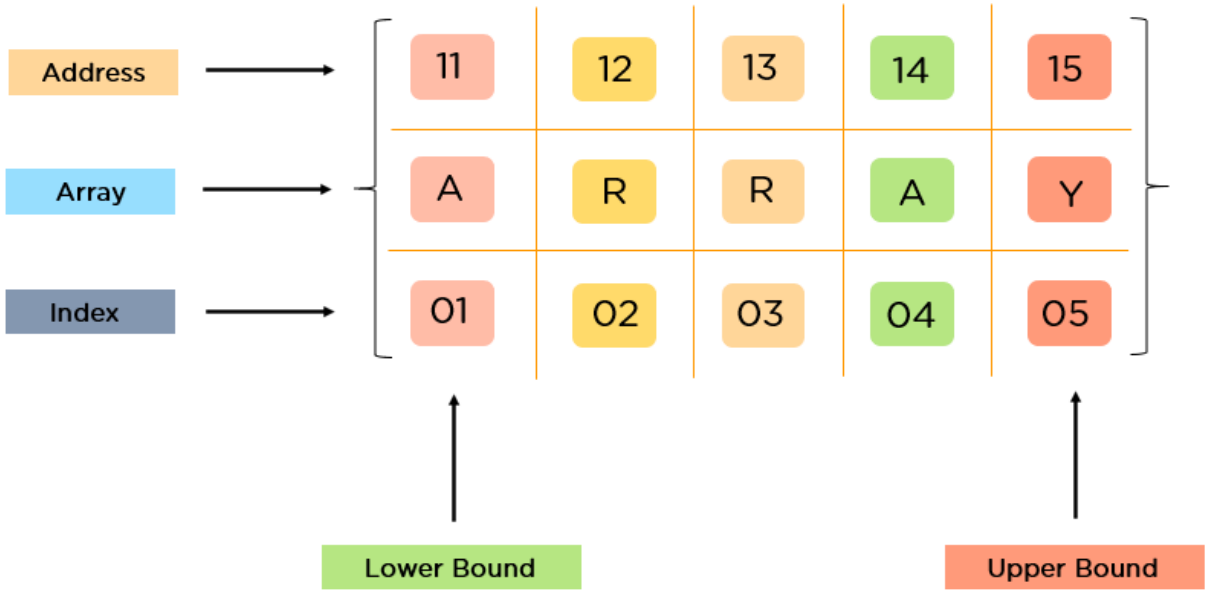
- **Applications of Recursion**

- **Writing base cases in Recursion**

- Factorial

- N-th Fibonacci number

ARRAYS



- **Introduction and Advantages**
- **Types of Arrays**

o Fixed-sized array

o Dynamic-sized array

- **Operations on Arrays o Searching**

o Insertions

o Deletion

o Arrays vs other DS

o Reversing - Explanation with complexity

SEARCHING

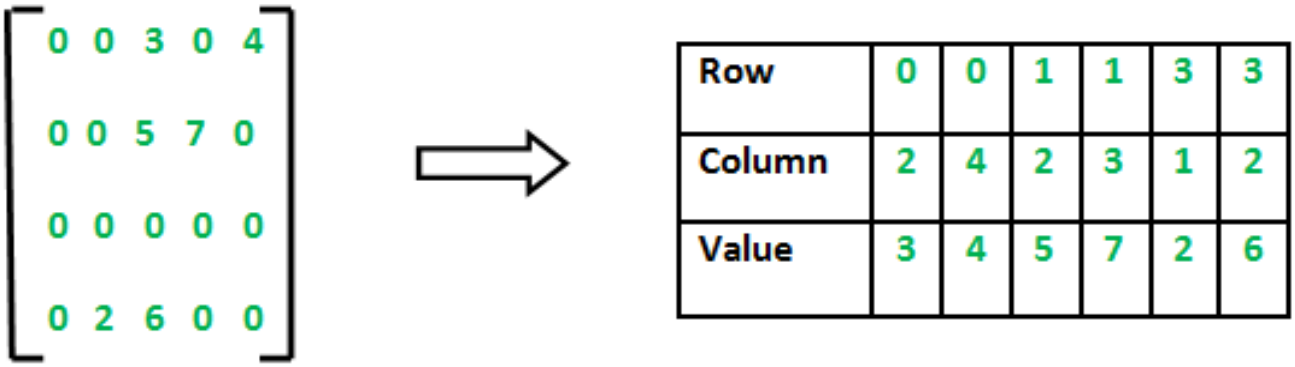
- **Binary Search Iterative and Recursive**
- **Binary Search and various associated problems**
- **Two Pointer Approach Problems**

SORTING

- **Implementation of C++ STL sort() function in Arrays and Vectors**
  - o Time Complexities
- **Sorting in Java**
- **Arrays.sort() in Java**
- **Collection.sort() in Java**
- **Stability in Sorting Algorithms**
  - o Examples of Stable and Unstable Algos
- **Insertion Sort**
- **Merge Sort**
- **Quick Sort**
  - o Using Lomuto and Hoare
  - o Time and Space analysis
  - o Choice of Pivot and Worst case

- **Overview of Sorting Algorithms**

MATRIX



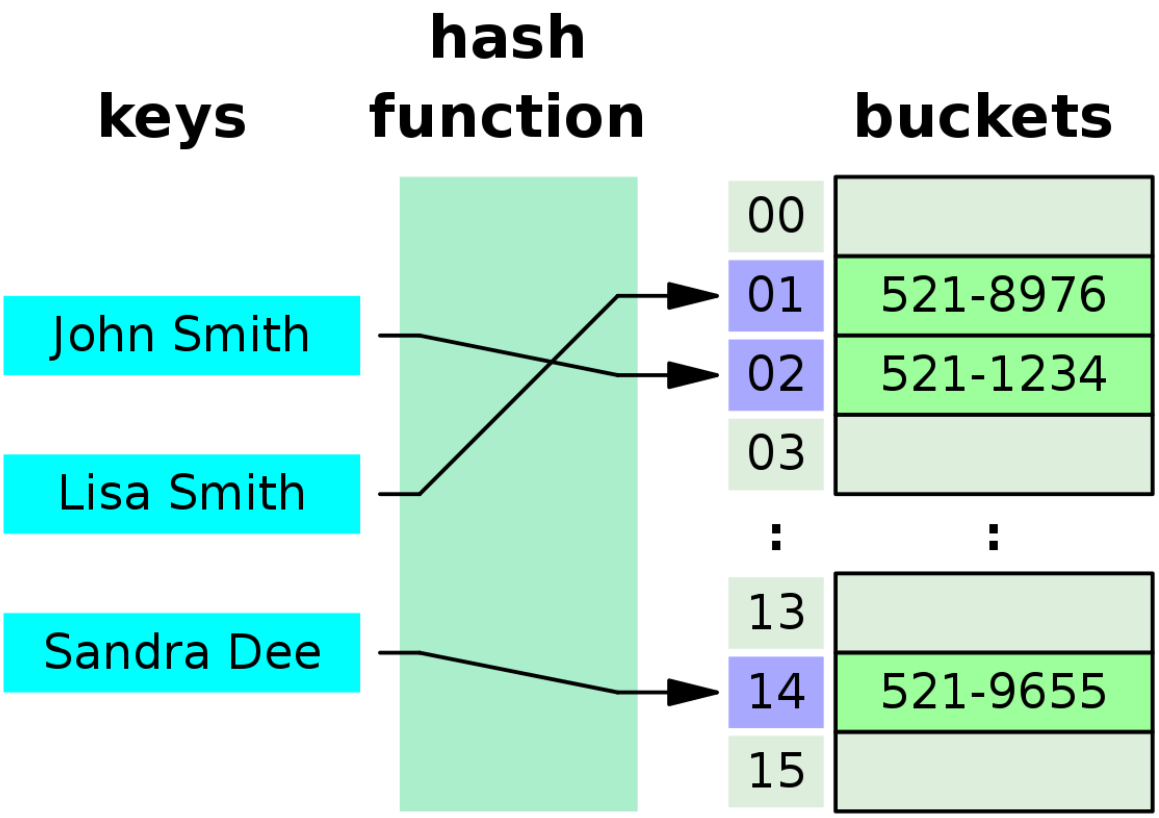
$$\begin{bmatrix} 0 & 0 & 3 & 0 & 4 \\ 0 & 0 & 5 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 6 & 0 & 0 \end{bmatrix}$$

Row	0	0	1	1	3	3
Column	2	4	2	3	1	2
Value	3	4	5	7	2	6

- **Introduction to Matrix in C++ and Java**
- **Multidimensional Matrix**
- **Pass Matrix as Argument**
- **Printing matrix in a snake pattern**
- **Transposing a matrix**
- **Rotating a Matrix**
- **Check if the element is present in a row and column-wise sorted matrix.**
- **Boundary Traversal**
- **Spiral Traversal**
- **Matrix Multiplication**

- **Search in row-wise and column-wise Sorted Matrix**

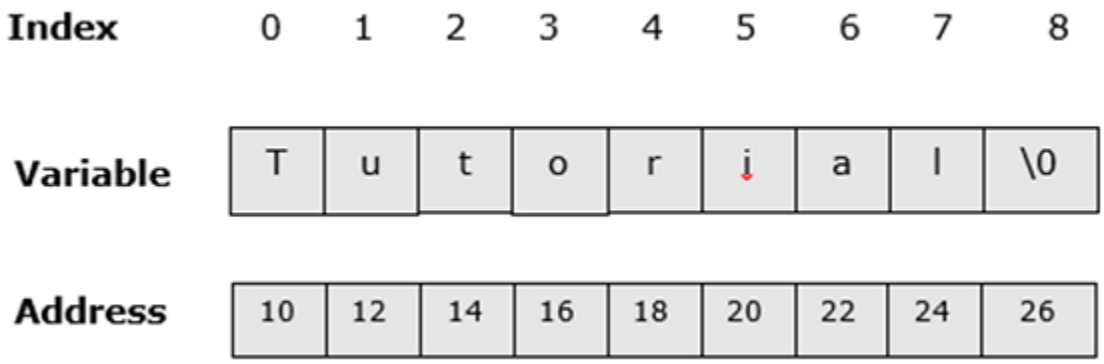
HASHING



- **Introduction and Time complexity analysis**
- **Application of Hashing**
- **Discussion on Direct Address Table**
- **Working and examples on various Hash Functions**
- **Introduction and Various techniques on Collision Handling**

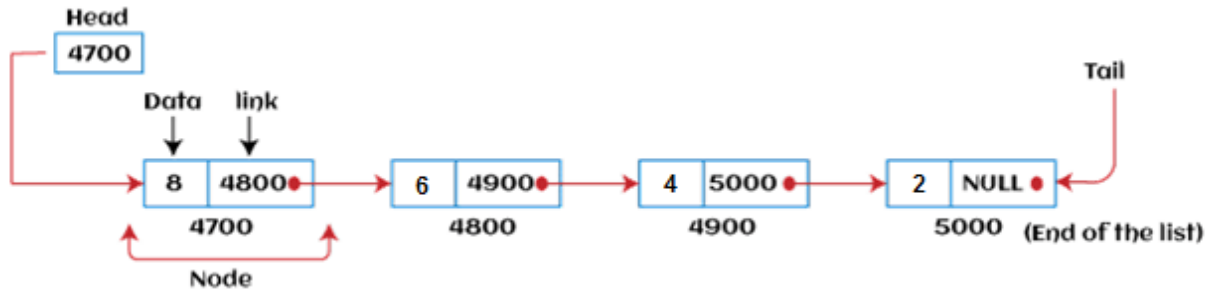
- **Chaining and its implementation**
- **Open Addressing and its Implementation**
- **Chaining V/S Open Addressing**
- **Double Hashing**
- **C++**
  - o Unordered Set
  - o Unordered Map
- **Java**
  - o HashSet
  - o HashMap

STRINGS



- **Discussion of String DS**
- **Strings in CPP**
- **Strings in Java**
- **Rabin Karp Algorithm**
- **KMP Algorithm**

LINKED LIST



- **Introduction**

o Implementation in CPP

o Implementation in Java

o Comparison with Array DS

- **Doubly Linked List**

- **Circular Linked List**

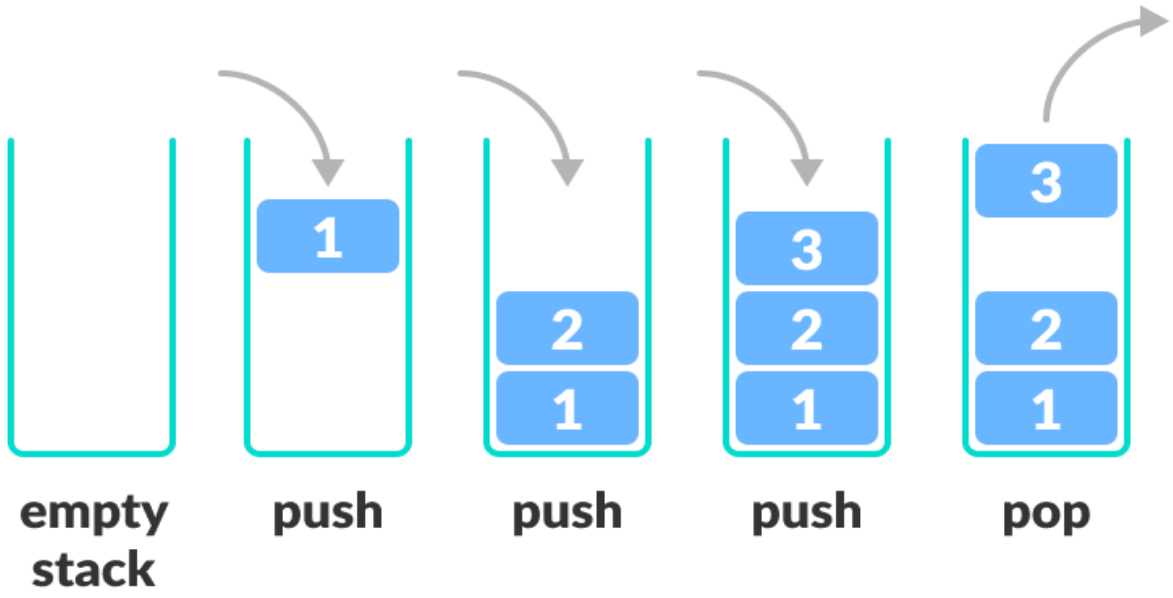
- **Loop Problems**

o Detecting Loops

o Detecting loops using Floyd cycle detection

o Detecting and Removing Loops in Linked List

STACK

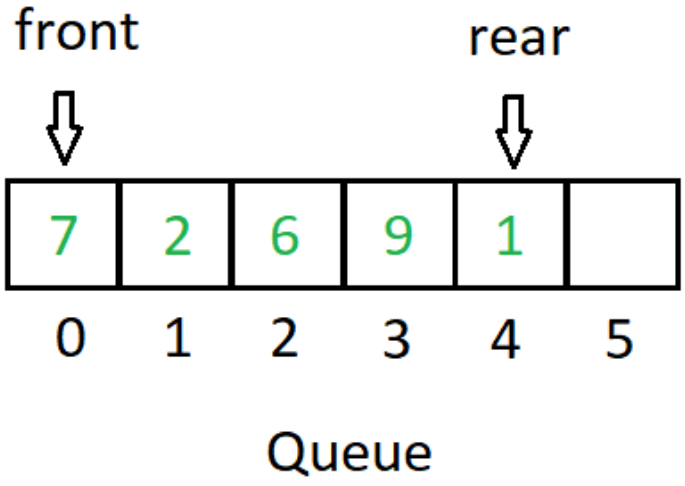


- **Applications of Stack**
- **Implementation of Stack in Array and Linked List**

o In C++

o In Java

QUEUE

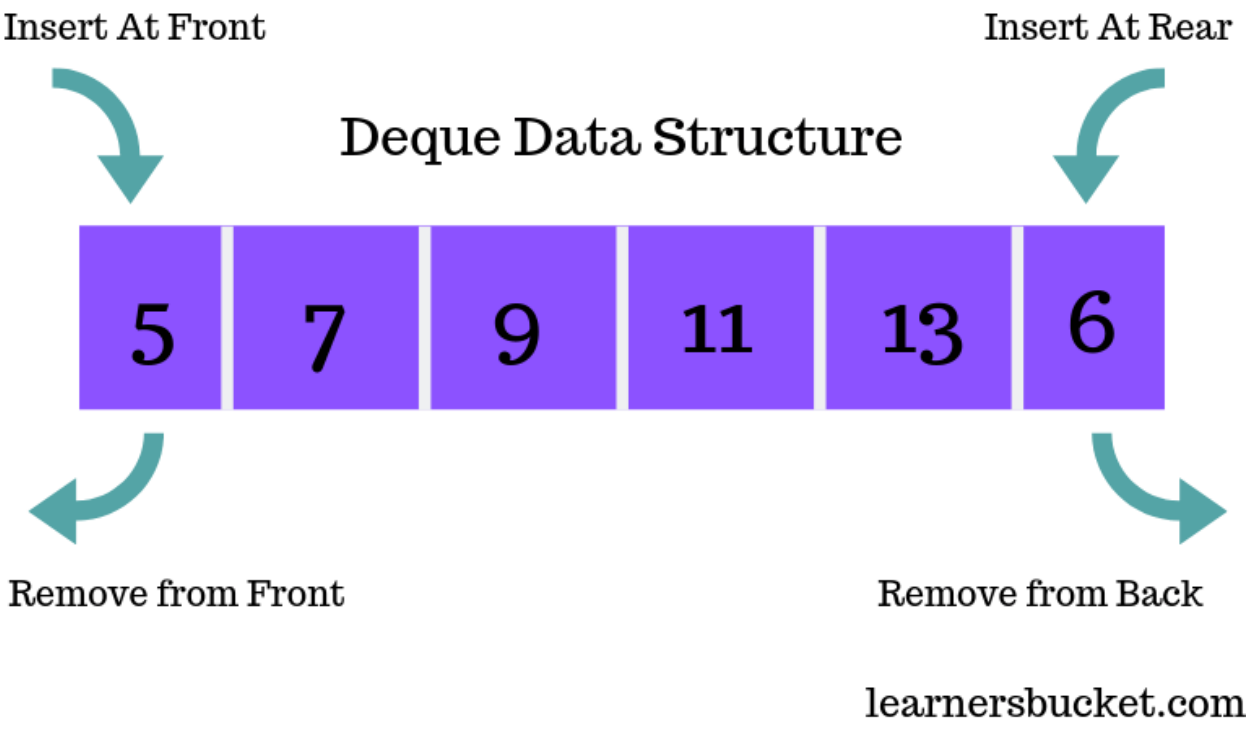


- **Introduction and Application**
- **Implementation of the queue using array and LinkedList**

o In C++ STL

o In Java o Stack using queue

DEQUE



- **Introduction and Application**
- **Implementation**

o In C++ STL

o In Java

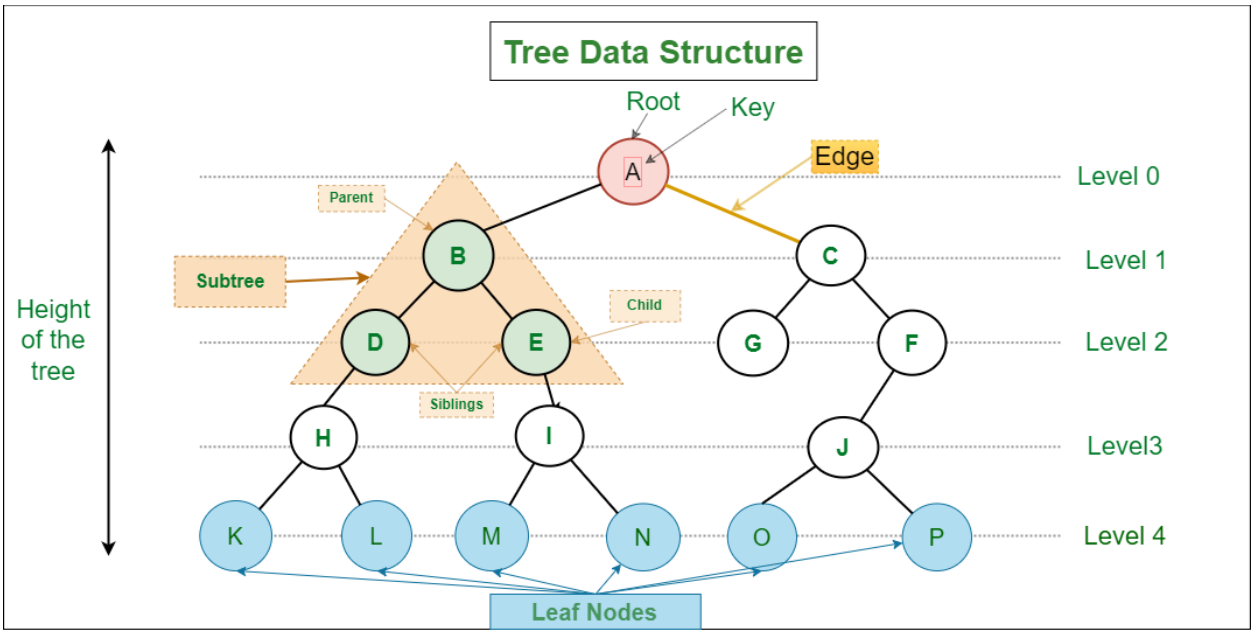
- **Problems(With Video Solutions)**

o Maximums of all subarrays of size k

o ArrayDeque in Java

o Design a DS with min max operations

TREES



- **Introduction**

o Tree

o Application

o Binary Tree

o Tree Traversal

- **Implementation of:**

o Inorder Traversal

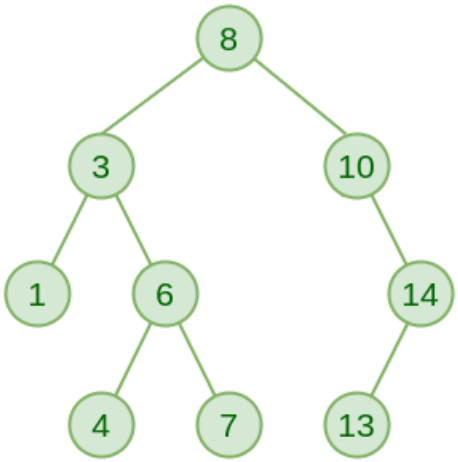
o Preorder Traversal

o Postorder Traversal

o Level Order Traversal (Line by Line)

o Tree Traversal in Spiral Form

BINARY SEARCH TREE



Binary Search Tree

- **Background, Introduction and Application**
- **Implementation of Search in BST**

o In CPP

o In Java

- **Insertion in BST**

o In CPP

o In Java

- Deletion in BST

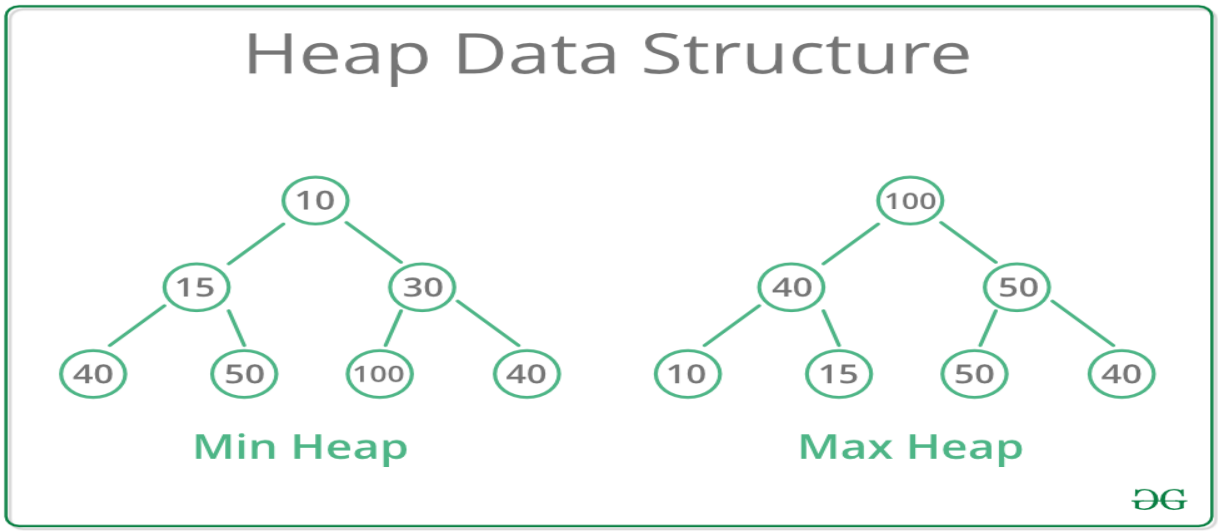
o In Java

- **Floor in BST**

o In Java

- **Self Balancing BST**
- **AVL Tree**
- **Red Black Tree**
- **Set in C++ STL**
- **Map in C++ STL**

HEAP



- **Introduction & Implementation**
- **Binary Heap**

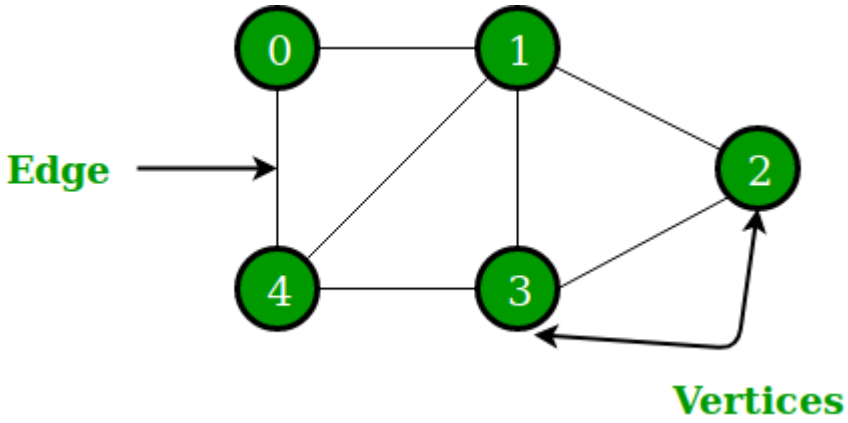
o Insertion

o Heapify and Extract

o Decrease Key, Delete and Build Heap

- **Heap Sort**
- **Priority Queue in C++**
- **PriorityQueue in Java**

GRAPH



- **Introduction to Graph**
- **Graph Representation**

o Adjacency Matrix

o Adjacency List in CPP and Java

o Adjacency Matrix VS List

- **Breadth-First Search**

o Applications

- **Depth First Search**

o Applications

- **Shortest Path in Directed Acyclic Graph**
- **Prim's Algorithm/Minimum Spanning Tree**

o Implementation in CPP

o Implementation in Java

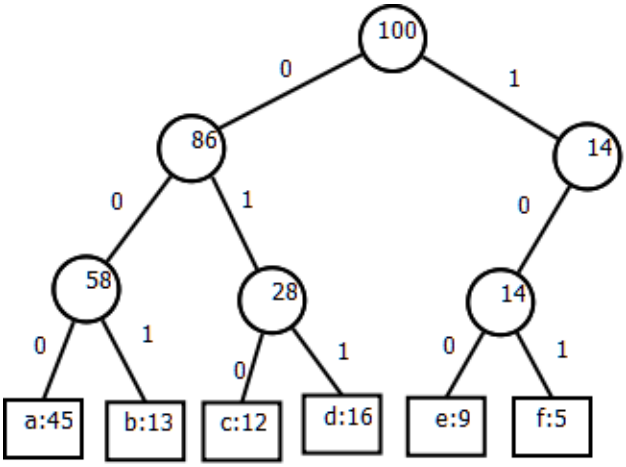
- **Dijkstra's Shortest Path Algorithm**

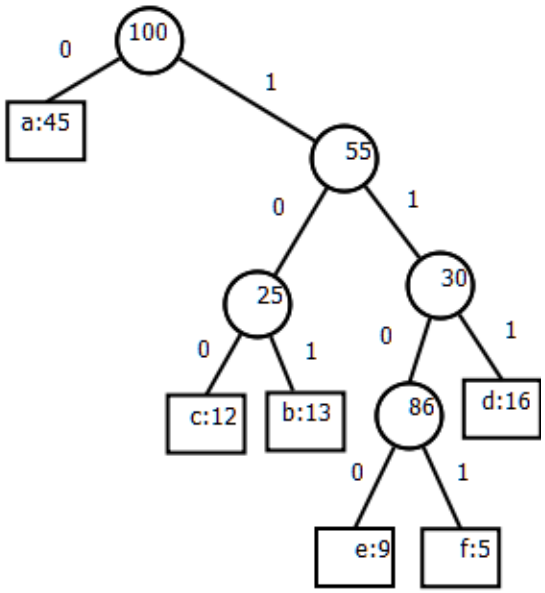
o Implementation in CPP

o Implementation in Java

- **Bellman-Ford Shortest Path Algorithm**
- **Kosaraju's Algorithm**
- **Articulation Point**
- **Bridges in Graph**
- **Tarjan’s Algorithm**

GREEDY





- **Introduction**
- **Activity Selection Problem**
- **Fractional Knapsack**
- **Job Sequencing Problem BACKTRACKING**
- **Concepts of Backtracking**
- **Rat In a Maze**
- **N Queen Problem**

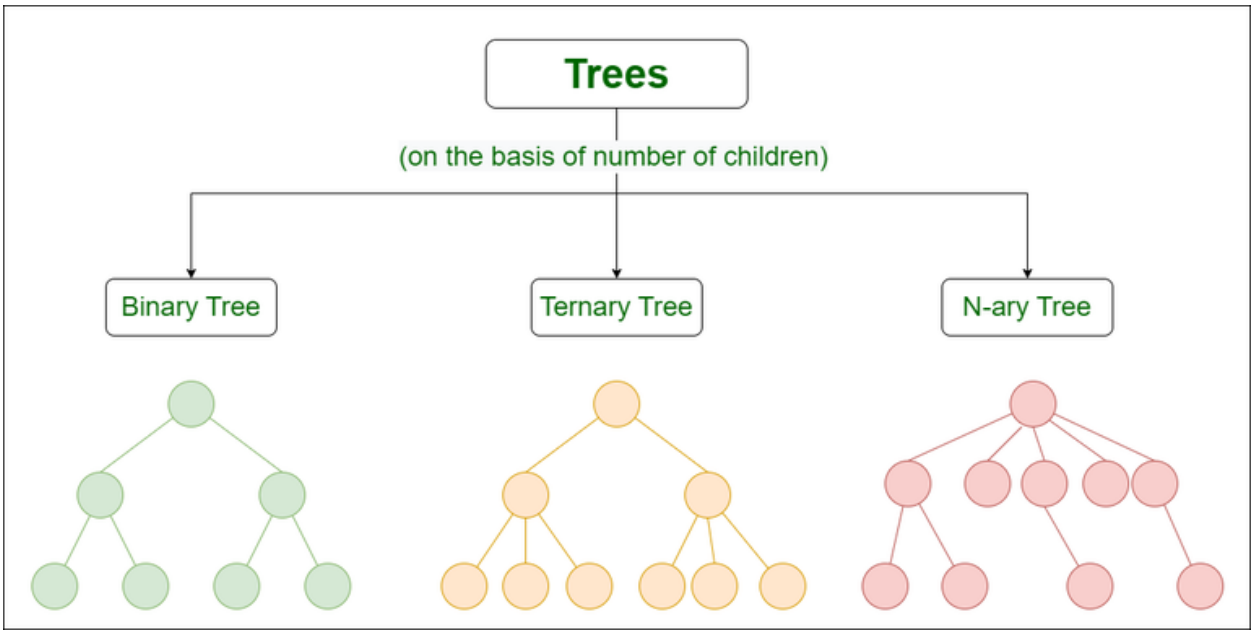
DYNAMIC PROGRAMMING

- **Introduction**
- **Dynamic Programming**

o Memoization

o Tabulation

TREE



- **Introduction**

o Representation

o Search

o Insert

o Delete

- **Count Distinct Rows in a Binary Matrix**

SEGMENT TREE

- **Introduction**
- **Construction**
- **Range Query**

- **Update Query**

DISJOINT SET

- **Introduction**
- **Find and Union Operations**
- **Union by Rank**
- **Path Compression**
- **Kruskal's Algorithm**

REASON FOR CHOOSING DSA

All of the above was part of my training during my summer break I specially choose the DSA by Geeks for Geeks for reasons stated below :

• I was interested in Problem Solving and Algorithms since my first semester.

• Data structure is a thing you need to know no matter in which language do you code.

• One need to learn how to make algorithm of a real life problem he/she is facing.

• It had video lectures of all the topics from which one can easily learn. I prefer learning from video rather than books and notes. I know books and notes and thesis have their own significance but still video lecture or face to face lectures make it easy to understand faster as we are involved Practically.

• It had 200+ algorithmic coding problems with video explaind solutions.

• It had track based learning and weekly assesment to test my skills.

• It contained a lot of knowledge for such a resonable price.

• The course was in two programing languages C++ and JAVA.

• This was a life time accessable course which I can use to learn even after my training whenever I want to revise.

• Along with all these reasons one of the reason was the Geeks for Geeks platform which is offering the course because Geeks for Geeks is one of the best platform for Computer Science Students.

LEARNING OUTCOMES

A lot of beginners and experienced programmers avoid learning Data Structures Algorithms and because it's complicated and they think that there is no use of all the above stuff in real life but there is a lot of implementation of DSA in daily life.

For example If we have to search our roll number in 2000 pages of Document how would we do that?

- If we try to search it randomly or in sequence it will take too much time.
- We can try another method in which we can directly go to page no. 1000 and we can see if our roll no. is there or not if not we can move ahead and by repeating this and eliminating we can search our roll no. in no time. And this is called Binary Search Algorithm.

Two reasons to Learn Data Structure and Algorithms –

- If you want to crack the interviews and get into the product based companies
- If you love to solve the real-world complex problems.

I have learnt a vast number of topics like Trees, Graphs, Linked Lists, Arrays, etc. I understood their basics, there working, there implementation, and their practical use in the problems we face while we solve a problem using coding.

When we work in IT sector (Software or Programing part to be specific) we need to solve the problems and make programs write tons of code which will help us with the given problem and to write a program one need to make different algorithms. Many algorithms combine to make a program. Now, algorithm are writen in some lenguages but they are not dependen ton them, one need to make a plan and algo first then write it into any language wether i tis C++ or JAVA or C or any other programing language. Algorith is based on data structure and its implementation and working. So, basiclly one need to have a good grip on DSA to work in programing sector.

When you ask someone to make a decision for something the good one will be able to tell you “I chose to do X because it’s better than A, B in these ways. I could have gone with C, but I felt this was a better choice because of this“. In our daily life, we always go with that person who can complete the task in a short amount of time with efficiency and using fewer resources. The same things happen with these companies. The problem faced by these companies is much harder and at a much larger

scale. Software developers also have to make the right decisions when it comes to solving the problems of these companies. Knowledge of data structures like Hash Tables, Trees, Tries, , and Graphs various goes a long way in solving these problems efficiently and the interviewers algorithms are more interested in seeing how candidates use these tools to solve a problem.

I learned about how to break a problem into pieces and then find the solution then how to maket he desired algorithm which will help me to solve my respective problem.

What I Learned from the course precisely :

• I Learned Data Structures and Algorithms from basic to advanced level.

• Learned Topic-wise implementation of different Data Structures & Algorithms.

• Improved my problem-solving skills to become a stronger developer.

• Developed my analytical skills on Data Structures and use them efficiently.

• Solved problems asked in product-based companies' interviews.

• Solved problems in contests similar to coding round for SDE role.

This will help me during my career as a programmer and afterwards also whenever I need to code. We are surrounded by a lot of real-world complex problems for which no one has the solution. Observe the

problems in-depth and you can help this world giving the solution which no one has given before.

“ Data structure and algorithms help in understanding the nature of the problem at a deeper level and thereby a better understanding of the world. ”

BIBLIOGRAPHY

- DSA Books
- Geeks for Geeks website
- Geeks for Geeks Course

----------THANK YOU----------