SIX WEEKS SUMMER TRAINING

REPORT

on

(JAVA & DSA / BOGGLE GAME SOLVER)

Submitted by

M. Lakshmi Narasa Reddy

Reg no: 11902757

P132: B. Tech (Computer Science & Engineering)

Under the Guidance of

Sehajpreet Singh

School of Computer Science & Engineering

Lovely Professional University, Phagwara

PHAGWARA, PUNJAB

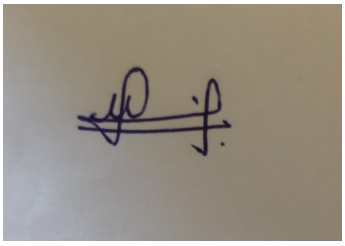
(June – July,2021)

DECLARATION

I hereby declare that I have completed my six-week summer training at **Cipher Schools** from

1st June 2021 to **14th July 2021** under the guidance of **Sehajpreet Singh.**

I have declared that I have worked with full dedication during these six weeks of training and my learning outcomes fulfil the requirements of training for the award of degree of **Computer Science and Engineering,** Lovely Professional University.



M. Lakshmi Narasa Reddy

Reg no:11902757

Date:

ACKNOWLEDGEMENT

I would like to express my gratitude towards my University as well as Geeks for Geeks for providing me the golden opportunity to do this wonderful summer training regarding DSA, which also helped me in doing a lot of homework and learning. As a result, I came to know about so many new things. So, I am really thank full to them.

Moreover I would like to thank my friends who helped me a lot whenever I got stuck in some problem related to my course. I am really thankfull to have such a good support of them as they always have my back whenever I need.

Also,I would like to mention the support system and consideration of my parents who have always been there in my life to make me choose right thing and oppose the wrong. Without them I could never had learned and became a person who I am now.

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

Summer Training completion Certificate by Cipher Schools

Certificate of Eminence

INTRODUCTION

JAVA & DSA course is a complete package that helped me to learn Data Structures and Algorithms from Basic to an Advance level. The course curriculm has been divided into 10 weeks, where I practiced questions and I have attempted the assesment tests accordingly. The course offers a wealth of programming challenges that helped me to learn all about DSA and making of an algorithm and how to solve problems and the logic behind the Algorithm.

The course was on live basis means I could join the lecture on time and interact with the lecturer if I have any doubts or the class continues. Content will be avilable to me always through Google classroom where the management uploads every lecture recording. There were live classes to learn, and questions given by the instructor to practice and also applied my learnings on competitive programming websites.

I learned Algorithmic techniques for solving various problems with full flexibility of time as I was not time bounded.

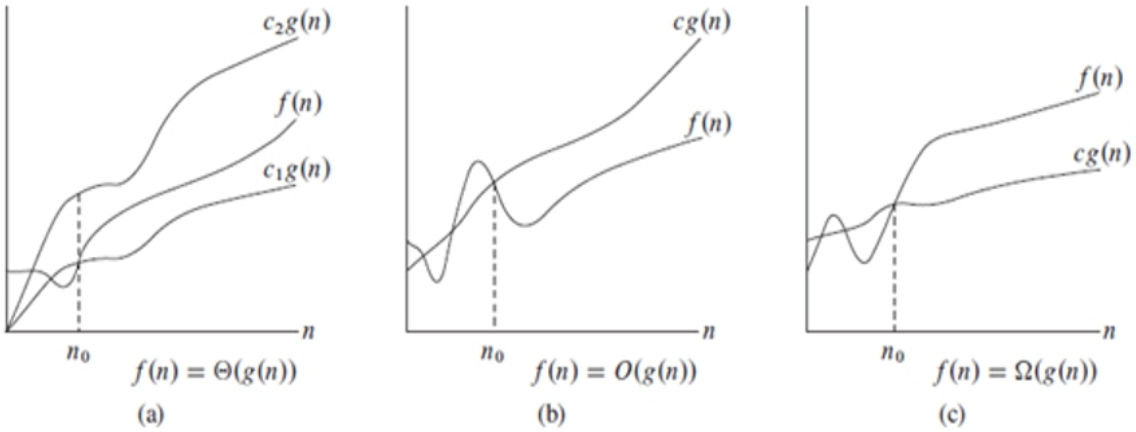
This course does not require any prior knowledge of Data Structure and Algorithms, but a basic knowledge of any programming language will be helpful.

And as we all know Data Structure and Algorithm is a must skill in terms of Placement in any company because it helps us to increase our problem-solving skill.

TECHNOLOGY LEARNT

INTRODUCTION TO DSA

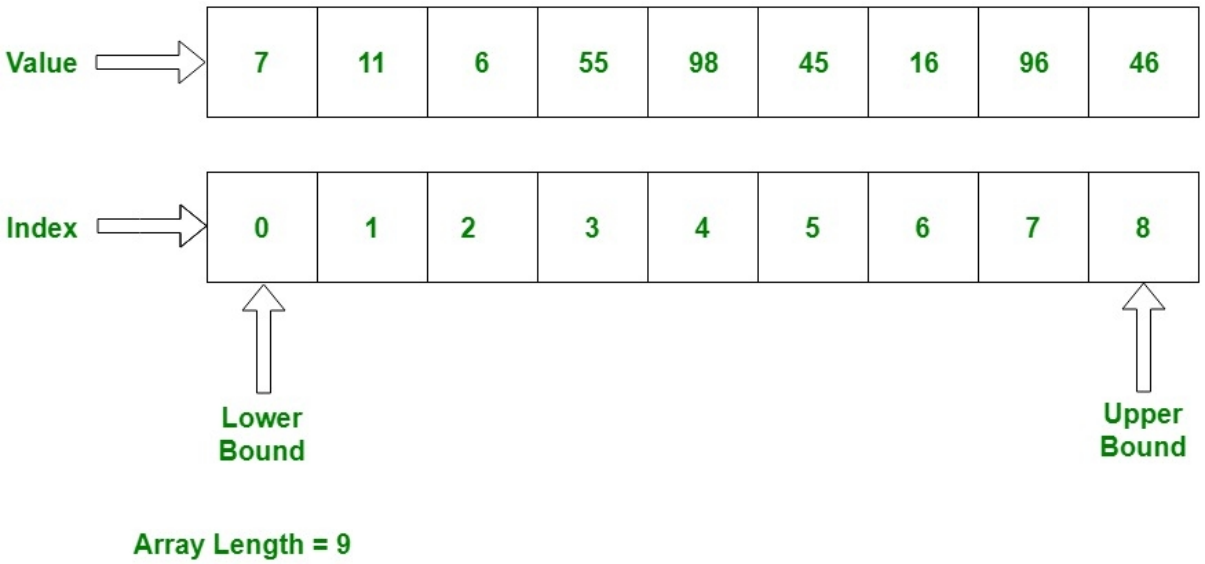
- **Analysis of Algorithm**
    - In this I learned about background analysis through a Program and its functions.
- **Order of Growth**
    - A mathematical explanation of the growth analysis through limits and functions.
    - A direct way of calculating the order of growth
- **Asymptotic Notations**
    - Best, Average and Worst case explanation through a program.
- **Big O Notation**
    - Graphical and mathematical explanation.
    - Calculation
    - Applications at Linear Search
- **Omega Notation**
    - Graphical and mathematical explanation.
    - Calculation.
- **Theta Notation**
    - Graphical and mathematical explanation.
    - Calculation.



- **Analysis of common loops**
    - Single, multiple, and nested loops
- **Analysis of Recursion**
    - Various calculations through Recursion Tree method
- **Space Complexity**
    - Basic Programs
    - Auxiliary Space
    - Space Analysis of Recursion
    - Space Analysis of Fibonacci number
- **Recursion**
    - Introduction to Recursion
    - Applications of Recursion
    - Stack overflow discussion
    - Ex: factorial, Nth Fibonacci number

ARRAYS

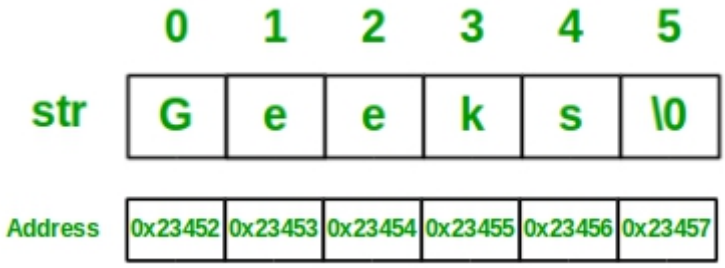
Array is an indexed collection of finite number of homogeneous data elements.



- **Introduction and Advantages**
- **Types of Arrays**
    - Fixed-sized array
    - Jagged Arrays
- **Operations on Arrays**
    - Searching
    - Insertions
    - Deletion
    - Arrays vs other DS
    - Reversing - Explanation with complexity
- **Techniques Implemented**
    - Sorting
    - Two Pointer Approach
    - Sliding Window Mechanism
    - ArrayList In Java

STRINGS

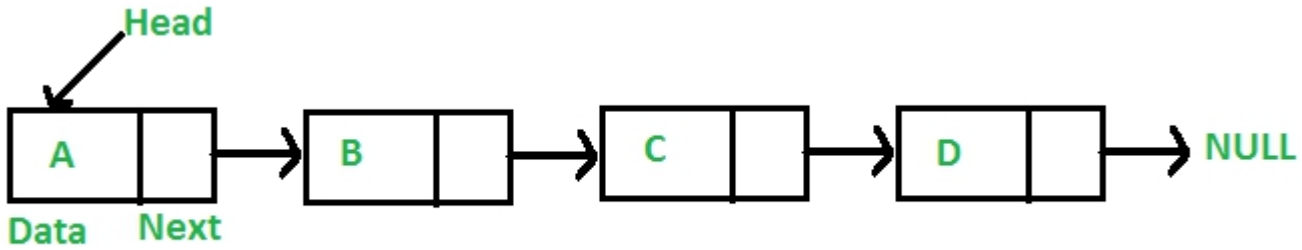
Strings in Java are Objects that are backed internally by a char array. Since arrays are immutable (cannot grow), Strings are immutable as well. Whenever a change to a String is made, an entirely new String is created.



- **String Discussions**
- **Strings in Java**
- **StringBuilder class In Java**
- **String Functions**
- **Sample Problems**
    - Anagrams
    - Palindrome
    - Reversing a String

LINKEDLIST

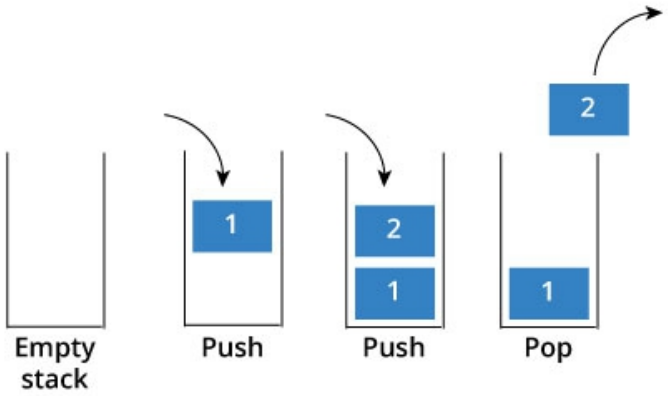
LinkedList is a linear data structure, where data is stored in the form of nodes and each node contains some data and link to next node.



- **Introduction**
    - Implementation in Java
    - Comparison with Array DS
- **Doubly Linked List**
- **Problems Solved**
    - Reversing a LinkedList
    - Intersection point in Y shaped LinkedList
    - Finding nth node from the end
- **Loop Problems**
    - Detecting Loops
    - Detecting loops using Floyd cycle detection
    - Detecting and Removing Loops in Linked List

STACKS

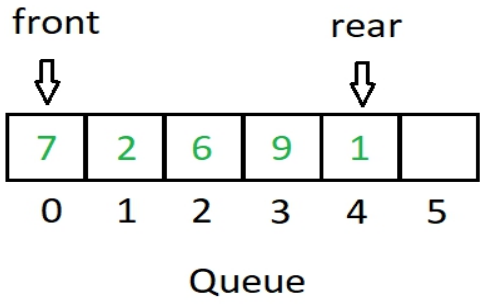
Stack is a linear data structure which follows a particular order in which the operations are performed. The order may be LIFO (Last in First Out) or FILO (First in Last Out).



- **Introduction**
- **Implementing with Array and LinkedList**
- **Problems Solved**
    - Balanced Parenthesis
    - Evaluate prefix/postfix questions
    - Infix to prefix/postfix conversion

QUEUE

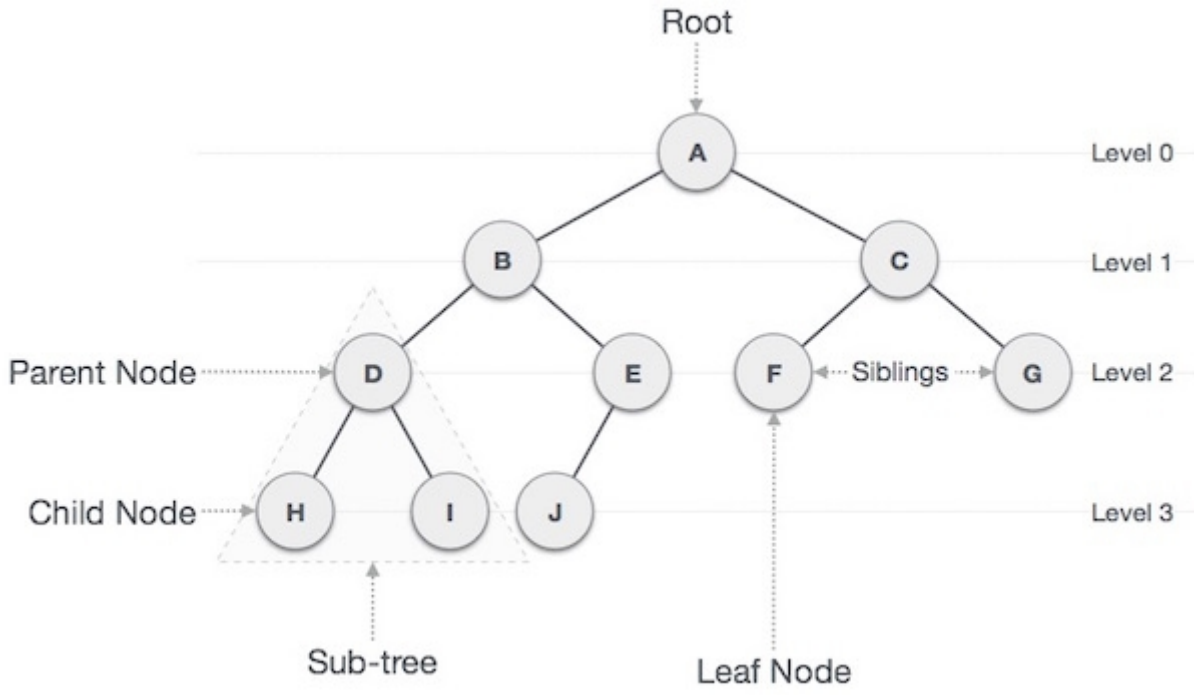
A Queue is a linear structure which follows a particular order in which the operations are performed. The order is First in First Out (FIFO)



- **Introduction**
- **Implementing with Arrays and LinkedList**
- **Types of Queues**

TREE

Tree is a nonlinear data structure stored in the form of nodes where each node contains some data and link to other nodes.



- **Introduction**
- **Different types of Trees**
- **Binary Search Tree**
    - A Binary Tree in which for each node, the value of nodes in left subtree is lesser/equal and value of nodes in right subtree are greater than the value of node.

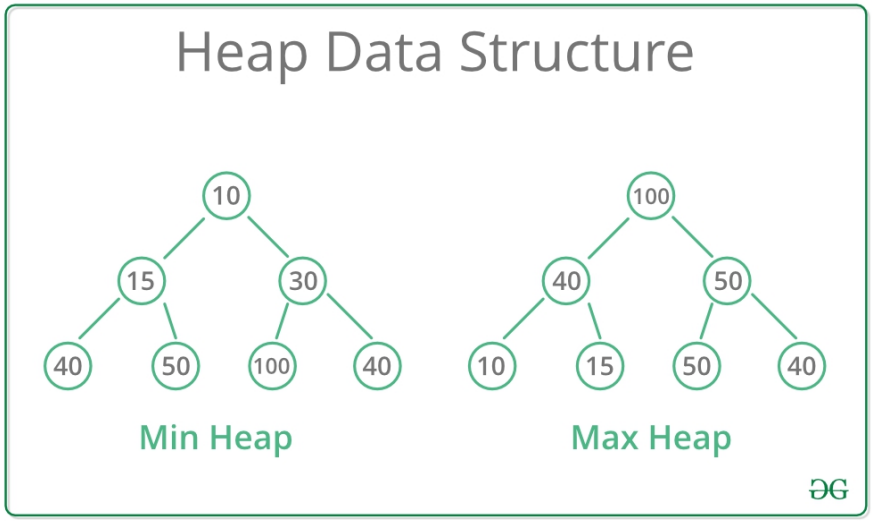
- **Implementation of BST**
- **Problems Solved**
    - Traversals (Bfs, Dfs)
    - Height of Binary tree
    - Counting nodes/Sum of all nodes
    - No of leaf nodes in a tree
    - Level order traversal
    - Lowest Common Ancestor

HEAP

A heap is a tree-based data structure in which all the nodes of the tree are in a specific order.

Two types:

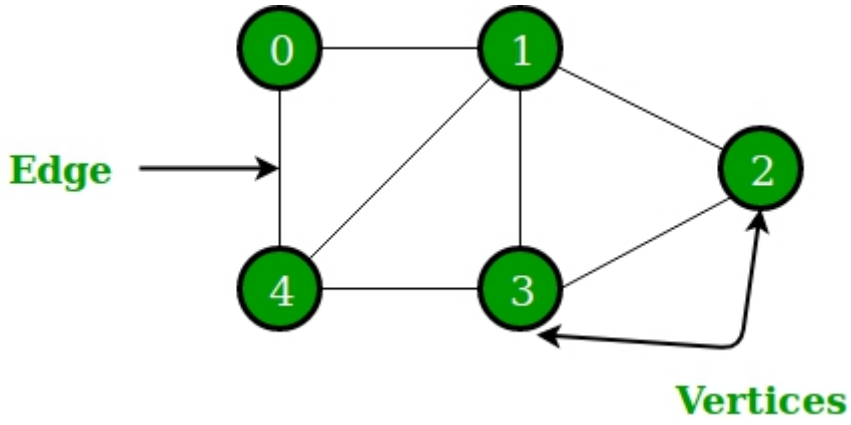
1. **Max Heap**
    - In a Max-Heap the key present at the root node must be greatest among the keys present at all its children.
2. **Min Heap**
    - In a Min-Heap the key present at the root node must be minimum among the keys present at all its children



- **Introduction**
- **Implementation**
- **Max Heap**
- **Min Heap**
- **Priority Queue**

GRAPH

A Graph is a non-linear data structure consisting of nodes and edges. The nodes are sometimes also referred to as vertices and the edges are lines or arcs that connect any two nodes in the graph.



The set of vertices V = {0,1,2,3,4} and the set of edges E = {01, 12, 23, 34, 04, 14, 13}.

- **Introduction**
- **Implementation**
  - Using Adjacency List Representation
  - Using Maps & Sets
- **Traversals**
  - Bfs and Dfs
- **Minimum Spanning Tree**
- **Dijkstra's Algorithm**
- **Prims Algorithm**
- **Kruskal's Algorithm**

PROJECT

Boggle game Solver

Boggle is a word game played using a plastic grid of lettered dice, in which players attempt to find words in sequence of adjacent letters.

Rules:

- You must find as many words as possible on the grid of at least length 3.
- You can move from one letter (dice) to another if it is a neighbour (in all eight directions).
- You cannot use a letter (dice) more than once in a word.
- You get points for each word - the more letters the better.

Implementation:

- Given a dictionary, we must use it
- So, we must import it somehow and store it
- Board is randomly generated.

- Traversing the array of Characters
- Check if that word is valid – that is it is there in our dictionary
- Check if a word is already found before If Valid, store

Algorithm:

- Input the Boggle Puzzle Board
- Input Dictionary to valid Data Structure (Trie)
- Calculate all possible words
- Show all valid words

Structure of Trie:

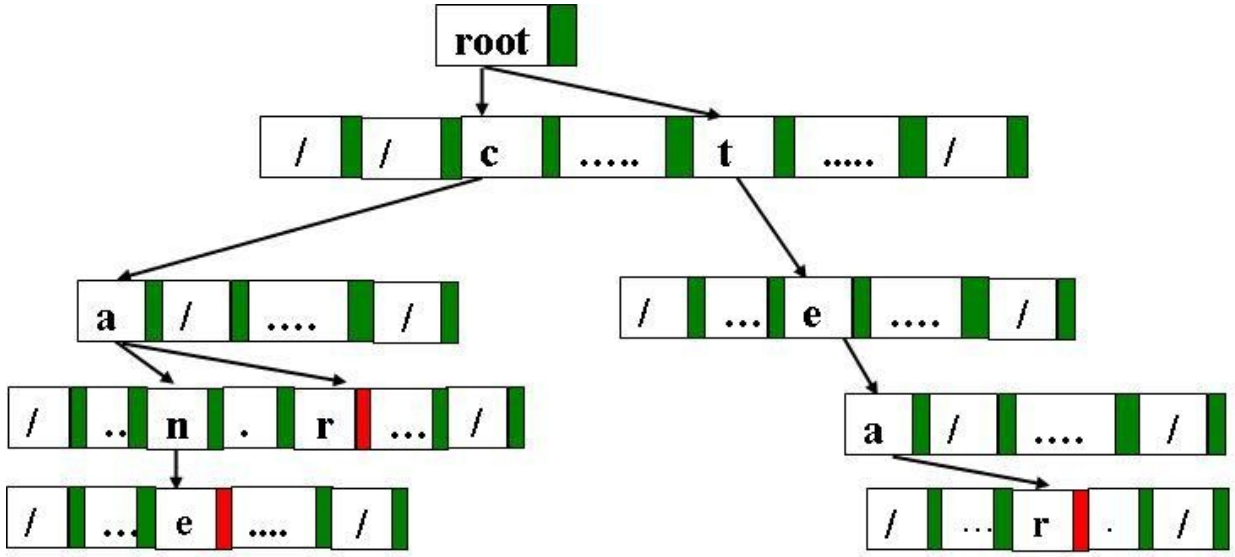


Figure 1: Trie Data Structure for words: cane, car, tear

Implementation of Boggle game Solver

Git hub link: [Boggle Game Solver]

REASONS FOR CHOOSING DSA

All of the above was part of my training during my summer break I specially choose the DSA by Cipher Schools 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 live lectures of all the topics from which one can easily learn. I prefer learning from Tutor rather than books and notes. I know books, notes and thesis have their own significance but still live face to face lectures makes it easy to understand faster as we are involved Practically.
- It had track based learning and weekly assesment to test my skills.
- It was a great opportunity for me to invest my time in learning instead of wasting it here and there during my summer break in this Covid-19 panademic.
- It contained a lot of knowledge for such a resonable price.
- The course was in JAVA.
- This was a life time accessable course which I can use to learn even after my training whenever I want to revise.

LEARNING OUTCOMES

A lot of beginners and experienced programmers avoid learning Data Structures and Algorithms 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 implementations 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 Structures 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 languages 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. Algorithm 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 decide 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 must make the right decisions when it comes to solving the problems of these companies.
Knowledge of data structures like Hash Tables, Trees, Tries, Graphs, and various algorithms goes a long way in solving these problems efficiently and the interviewers 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 Programming Platforms like Hacker Rank, Code chef and geeks for geeks.

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."

BIBILIOGRAPHY

- Cipher Schools
- Geeks for Geeks
- Hacker Rank
- Coding Simplified.com