

# **SIX WEEKS SUMMER TRAINING**

**Report on**

## **DSA Self- paced Course by GFG**

**Submitted to**

**LOVELY PROFESSIONAL UNIVERSITY**

**PHAGWARA, PUNJAB**



**From: June 2022 to July 2022**

**Submitted By:**

**Name of student: Rohan Raj**

**Registration Number: 12017259**

## **DECLARATION**

I hereby declare that I have completed my six weeks summer course from GeeksforGeeks from June 2022 to July 2022. I have declared that I have learned with full dedication during these six weeks of course and my learning outcomes fulfil the requirements of training for the award of degree of DSA self-placed course.

Lovely Professional University, Phagwara.

Name – Rohan Raj

Reg No - 12017259

Date: 23 September 2022

# CERTIFICATE



# CERTIFICATE

## OF COURSE COMPLETION

THIS IS TO CERTIFY THAT

**Rohan Raj**

has successfully completed the course on DSA Self paced of duration 8 weeks.

*Sandeep Jain*

**Mr. Sandeep Jain**  
Founder & CEO, GeeksforGeeks

<https://media.geeksforgeeks.org/courses/certificates/a00dc180c6561262570985ecb1b28311.pdf>

# **Table of Contents:**

-  Introduction
-  Asymptotic Analysis
-  Mathematics
-  Bit magic
-  Recursion
-  Array
-  Searching
-  Sorting
-  Matrix
-  Hashing
-  Strings
-  Linked List
-  Stack
-  Queue



## **INTRODUCTION**

This course is a complete package for learning all the concepts of DSA. This course that has been divided into 8 weeks where we can learn the basics of DSA and can practice questions & attempt the assessment tests from anywhere in the world. This will further help us to prepare for interviews with top-notch companies like Microsoft, Amazon, and Adobe etc. We have also learned algorithmic techniques for solving various problems with full flexibility of time. This course does not require any prior knowledge of Data Structure and Algorithms, but a basic knowledge of any programming language (C++ / Java).

This course has so many questions for practice after every track also it consist so many contests by solving this we can practice for Big companies contests and also it has questions to solve which were asked previously by some another companies and also they provide doubt support If you gets sucked in any problem they help us to how can we solve that problem by providing some hints.

In this course we can learn DSA from both the programming language Java. There are particular videos for both the languages you can choose which language we want to choose. I have completed this course from Java.

## **Course Goals and Objectives**

Mastering DSA from basic to advanced level

Solving problems which are asked in product-based companies

Solve problems in contests similar to coding round for SDE role

Become a strong and efficient developer

Practice for Contests



# **Introduction to Company GeeksForGeeks**

It's a computer science portal for geeks, by geeks. GeeksforGeeks.org was created with a goal in mind to provide well written, well thought and well explained solutions for selected questions. The core team of five super geeks constituting of technology lovers and computer science enthusiasts have been constantly working in this direction.

The content on GeeksforGeeks has been divided into various categories to make it easily accessible for the users. Whether you want to learn algorithms, data structures or it is the programming language on its own which interests you, GeeksforGeeks has covered everything! Even if you are looking for Interview preparation material, GeeksforGeeks has a vast set of company-wise interview experiences to learn from, that gives user insights into the recruitment procedure of a company. Adding to this, it serves as a perfect platform for users to share their knowledge via contributes option.

GeeksforGeeks realizes the importance of programming practice in climbing the stairs of success in the field of Computer Science. That is why, it also provides an option of practicing problems. This huge database of problems is created by programming experts. The active team of GeeksforGeeks makes the learning process interesting and fun.

A large portion of the fun of programming comes because of the cool and interesting projects. You want to try out some Image Processing or Computer Vision projects, or if you want to make your own small game, GeeksforGeeks has covered it all for you. Just go ahead and search your favorite project and with the help of the code and explanation provided by the author, start your journey to a nice and cool project!

Last but not the least, GeeksforGeeks believes in the power of experience. That is why, it interviews pro-coders and shares those words of wisdom with everyone, giving them a sort of standard operating procedure which any aspirant can follow to achieve the same (and even better) level of success.

# Asymptotic Analysis

- **Analysis of Algorithm**
  - a) Background analysis through a Program and its functions.
- **Order of Growth**
  - a) A mathematical explanation of the growth analysis through limits and functions.
  - b) 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
- **Practice Problems**
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.



# **Mathematics**

- **Mathematics**

- Count Digits
- Palindrome Numbers
- Factorial of Numbers
- GCD of Two Numbers
- LCM of Two Numbers
- Check for Prime
- Prime Factors
- Sieve of Eratosthenes
- Computing Power

- **Practice Problems**

- This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

## **Bit magic**

- **Bitwise Operators in Java**

- Operation of AND, OR
- Operation of Bitwise Not, Left Shift
- Operation of Right Shift and unsigned Right Shift

○

- **Problem: Check Kth bit is set or not**

- Method 1: Using the left Shift.
- Method 2: Using the right shift

- **Problem: Count Set Bits**

- Method 1: Simple method
- Method 2: Brian and Kernighan's Algorithm
- Method 3: Using Lookup Table

- **Problems:**

- To check whether a number is a power of 2 or not
- Odd occurrences in an array.
- Two numbers having odd occurrences in an array.
- Generate power set using bitwise operators.

- **Practice Problems**

- This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

# **Recursion**

- Introduction to Recursion
- Applications of Recursion
- Writing base cases in Recursion
  - Factorial
  - N-th Fibonacci number
- Various problems on Recursion(With Video Solutions)
  - Print n to 1
  - Print 1 to n
  - Tail Recursion
  - Checking Palindrome
  - Sum of digits
  - Rod cutting
  - Subsets of a set
  - Tower of Hanoi Problem
  - Josephus Problem
- Practice Problems
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

# Array

- Introduction and Advantages
- Types of Arrays
  - Fixed-sized array
  - Dynamic-sized array
- Operations on Arrays
  - Searching
  - Insertions
  - Deletion
  - Arrays vs other DS
  - Reversing - Explanation with complexity
- Problems(With Video Solutions)
  - Left Rotation of the array by 1
  - Check if Sorted
  - Left Rotation of the array by D places
  - Leaders in an Array
  - Maximum Difference Problem
  - Frequencies in Sorted Array
  - Stock Buy and Sell Problem
  - Trapping Rainwater Problem
  - Maximum Consecutive 1s
  - Maximum Subarray Sum
  - Longest Even-Odd Subarray
  - Maximum Circular sum subarray.
  - Majority Element
  - Minimum Consecutive Flips
  - Sliding Window Technique
  - Prefix Sum Technique
- Practice Problems

This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

# Searching

- Binary Search Iterative and Recursive
- Binary Search and various associated problems
  - Index of First Occurrence in Sorted Array
  - Index of Last Occurrence in Sorted Array
  - Count of occurrences of x in sorted element
  - Count of 1s in a binary sorted array
  - Find an element in sorted and rotated array
  - Peak element
  - Find an element in an infinite sized sorted array
  - The square root of an integer
- Two Pointer Approach Problems(With Video Solutions)
  - Find pair in an unsorted array which gives sum X
  - Find pair in a sorted array which gives sum X
  - Find triplet in an array which gives sum X
- Problems(With Video Solutions)
  - Median of two sorted arrays
  - Majority Element
- Practice Problems
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

# Sorting

- Sorting in Java
- Arrays.sort() in Java
- Collection.sort() in Java
- Stability in Sorting Algorithms
- Bubble Sort
- Selection Sort
- Insertion Sort
- Merge Sort
- Problems(With Video Solutions)
  - Intersection of 2 sorted arrays
  - Union of 2 sorted arrays
  - Count Inversions in arrays
- Partitions(With Video Solutions)
  - Naive
  - Lomuto
  - Hoare
- Quick Sort
  - Using Lomuto and Hoare
  - Time and Space analysis
  - Choice of Pivot and Worst case
  - Tail call elimination
- Problems(With Video Solutions)
  - Kth Smallest element
  - Chocolate Distribution Problem
  - Sorting arrays with 2 and 3 types of elements
  - Merge Overlapping Intervals
  - Meeting the Maximum Guests
- Heap Sort
- Cycle Sort
- Counting Sort
- Radix Sort
- Bucket Sort

# **Matrix**

- Introduction to Matrix in 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
- **Java**
  - HashSet
  - HashMap
- **Problems(With Video Solutions):**
  - Count Distinct Elements
  - Count of the frequency of array elements
  - The intersection of two arrays
  - Union of two unsorted arrays
  - Pair with given sum in an unsorted array
  - Subarray with zero-sum
  - Subarray with given sum
  - Longest subarray with a given sum
  - Longest subarray with an equal number of 0's and 1's
  - Longest common span with the same sum in a binary array
  - Longest Consecutive Subsequence
  - Count Distinct elements in every window
  - More than n/k Occurrences
  - Optimized More than n/k Solution



# Strings

- Discussion of String DS
- Strings in CPP
- Strings in Java
- **Problems(With Video Solutions):**
  - Given a string, check if they are an anagram of each other.
  - Given a string, find the leftmost character that repeats.
  - Given a string, find the leftmost character that does not repeat.
  - Given a string, find the lexicographic rank of it in  $O(n)$  time.
  - Implementation of the previously discussed lexicographic rank problem.
  - Given a text string and a pattern string, find if a permutation of the pattern exists in the text.
  - Given two strings, check if they are rotations of each other or not.
  - Various Pattern Searching Algorithms.
  - Palindrome Check
- Rabin Karp Algorithm
- KMP Algorithm
- Practice Problems
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

# **Linked List**

- **Introduction**
  - Implementation in CPP
  - Implementation in Java
  - Comparison with Array DS
- **Doubly Linked List**
- **Circular Linked List**
- **Loop Problems**
  - Detecting Loops
  - Detecting loops using Floyd cycle detection
  - Detecting and Removing Loops in Linked List
- **Practice Problems**
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

# **Stack**

- Understanding the Stack data structure
- Applications of Stack
- Implementation of Stack in Array and Linked List
- Problems(With Video Solutions):
  - Balanced Parenthesis
  - Two stacks in an array
  - K Stacks in an array
  - Stock span problem with variations
  - Previous Greater Element
  - Next Greater Element
  - Largest Rectangular Area in a Histogram
- Understanding getMin() in Stack with  $O(1)$
- Infix, Prefix and Postfix Introduction

# Queue

- Introduction and Application
- Implementation of the queue using array and LinkedList
  - In Java
  - Stack using queue
- Problems(With Video Solutions)
  - Reversing a Queue
  - Generate numbers with given digits
  - First Circular Tour
- Practice Problems
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.



## **CONCLUSION**

After the completion of this course I have learned more depth concepts of DSA also solved many practice problems which will help me in solving any programming questions. This course also helped me to analyse the time complexities of various codes and also how much space any program takes to execute. and how can we implement that with their inbuilt functions.

## Sudoku Solver:

Sudoku is a 2D grid game, so this data structure comes in mind intuitively.

Mostly the operation involves reading the value of cell and writing value to the cell. So, these operations must be as fast as possible. With the use of 2D array, these operations will be  $O(1)$ . So this is the data structure which we used.

### *The algorithm used:-*

This is problem that is very appropriate for backtracking, as a digit in the wrong location often quickly shows that the solution is infeasible.

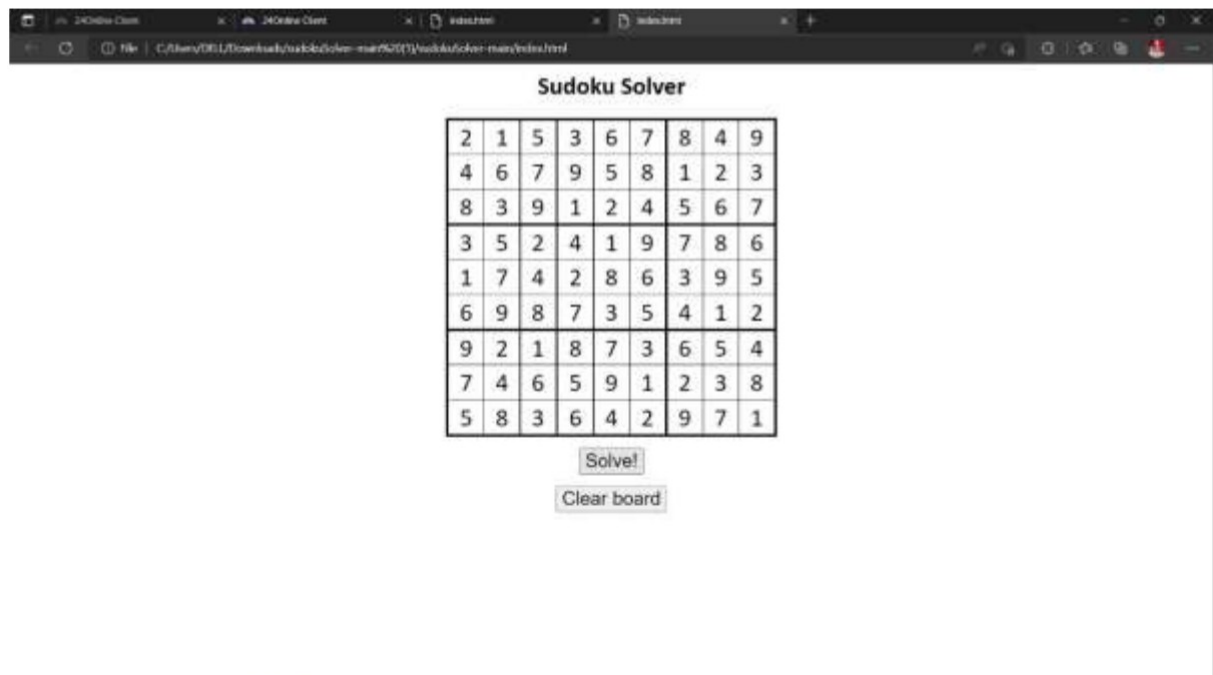
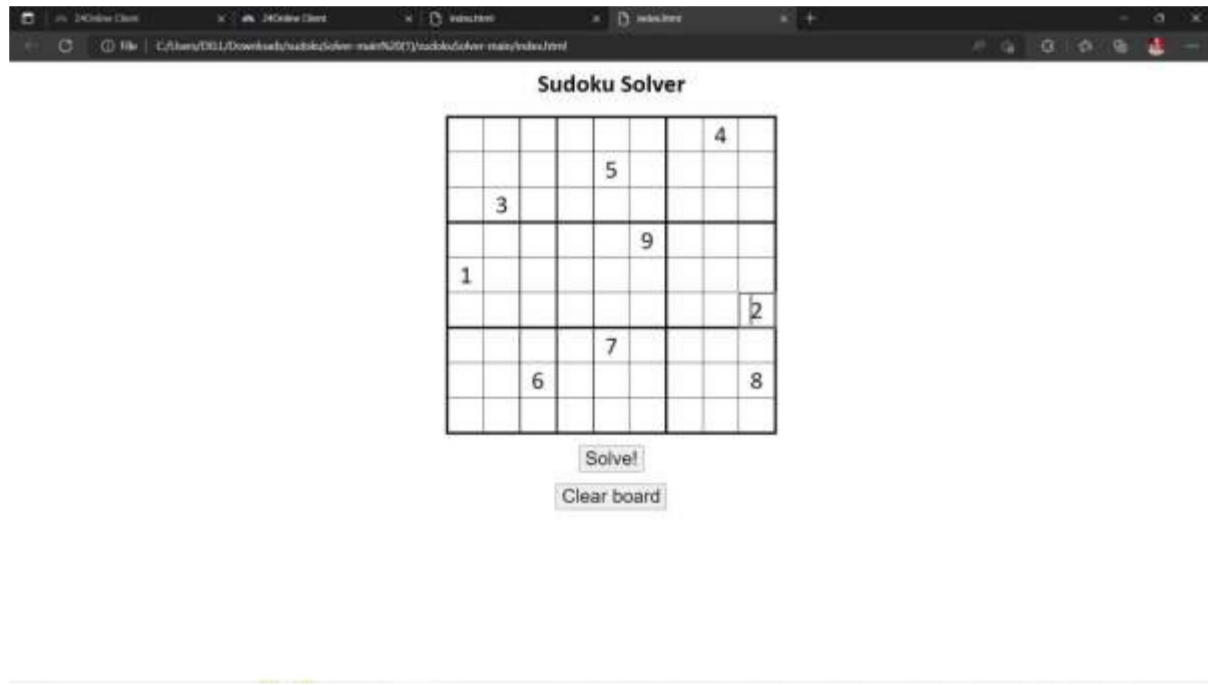
With  $n$  entries left,

- if there are no entries,  $n = 0$ , left, we are finished, indicate success;
- otherwise, find a square that is not yet filled, and
- for each digit from 1 to 9, ◦ place the digit in the digit in that square and see whether the solution is feasible, and if so call backtracking
  - algorithm recursively, where
    - ← if the algorithm indicates success, we are finished,
    - ← otherwise, try the next digit; and
- if no digit works, there is no solution.

This algorithm is implemented in the source directory

**Also, HTML is used for giving it as a look of a website where we can play the game, DSA is applied with JavaScript.**

## Project Outlook:-



## **Learning Outcome from training/technology learnt**

- Basics of data structures
- Analysis of algorithm
- Arrays
- Strings/queue/stack
- Sorting
- Insertion
- Graphs
- Trees
- Heap
- Hashing
- Time complexities
- Space complexities