

- Course Logistics 1/1 Lectures Completed

About Course Assignments and Doubts

- Programming Fundamentals 1.0 (Basics) 14/14 Lectures Completed

CPP - Understanding Computers, Programming Languages & Algorithms

CPP - Data and Variables

CPP - Basic Programming Instructions

CPP ★ Installation

CPP ★ Hello World

CPP - Basic c++ code

CPP - Basic c++ Concepts

CPP - Datatypes and Storage

CPP - Avg of Numbers/ Implicit Type Conversion

CPP - Type Conversion

CPP - Simple Interest

CPP Data Modifiers

Puzzle Hour Glass

Quiz On Programming Fundamentals 1

- Programming Fundamentals 2.0 (Loops & Patterns) 13/13 Lectures Completed

Loops - While Loop

While Loop - Sum of Even Numbers

CPP - Reading N Inputs

CPP - Break and Continue

Problem Submissions Leaderboard

```
1 #include<iostream>
2 int main() {
3 |     return 0;
4 }
```

SIMPLE INPUT

Given a list of numbers, stop processing input after the cumulative sum of all the input becomes negative.

Input Format:

A list of integers to be processed

Constraints:

All numbers input are integers between -1000 and 1000.

Output Format

Print all the numbers before the cumulative sum become negative.

C++ ▼ Download Testcases Unlock Editorial Submit Code

Custom Input Run Code Output :

- EPI - Break and Continue

- ▶ CPP - Basic Prime Check ✓
- ▶ CPP - Loops - Sum of a digits of a number ✓
- ▶ CPP - Loops - For loop ✓
- ▶ CPP - Loops - Fibonacci sequence ✓
- ▶ CPP - Loops - Number pyramids ✓
- ▶ CPP - Loops - Number Pattern ✓
- ▶ CPP - Loops - ABCD Pyramid Pattern ✓
- ▶ CPP - Loops - Square Root ✓
- ▶ Puzzle - Infinite quaters ✓

- Programming Fundamentals 3.0 (Operators)

12/12 Lectures Completed

- ▶ CPP - Size of Operator 6110 ✓
- ▶ CPP - Ternary Operator ✓
- ▶ CPP - Bitwise Operators ✓
- ▶ CPP - Loops - Finding Setbits ✓
- ▶ CPP - Loops - Ones Zeroes Pattern (Bitwise, Ternary Operator) ✓
- ▶ CPP - Operators Post and Pre-increment ✓
- ▶ CPP - Operator Associativity and Precedence ✓
- ▶ CPP - Scope of Variable ✓
- ▶ CPP - cin vs cin.get ✓
- ▶ CPP - Shortest Road Trip ✓
- ▶ CPP - Reading Till End of File ✓
- ▶ CPP - Apples and Oranges ✓

- Challenges - Fundamentals

8/18 Lectures Completed

- </> 💡 Simple Input ●
- </> 💡 Revising Quadratic Equations ●
- </> 💡 Print reverse ✓
- </> 💡 Von Neuman Loves Binary ✓
- </> 💡 Binary To Decimal ✓
- </> 💡 Decimal To Octal ●
- </> 💡 Prateek Loves Candy ●
- </> 💡 Print Series ✓
- </> 💡 Delhi's Odd Even ●
- </> 💡 Conversion (Fahrenheit to Celsius) ●
- </> 💡 Check prime ✓
- </> 💡 Count Digits ✓
- </> 💡 Increasing Decreasing Sequence ●
- </> 💡 Is Armstrong Number ✓
- </> 💡 Basic Calculator ✓
- </> 💡 Help Ramu ●
- </> 💡 Pythagoras Triplet ●
- ▶ Pythagorean Triplet Hint ●

- 🚀 Challenges - Patterns 5/15 Lectures Completed
- </> 💡 Fibonacci Pattern (Pattern 4) ✓
 - </> 💡 Manmohan Loves Patterns - I ✓
 - </> 💡 Manmohan Loves Patterns- II ✓
 - </> 💡 Pattern Mountain ●
 - </> 💡 Pattern with Zeros ✓
 - </> 💡 Pattern Triangle ●
 - </> 💡 Pattern DoubleSidedArrow ●

Pattern DoubleSidedArrow		
	Puzzle Hour Glass	
	💡 Pattern InvertedHourGlass	
	💡 ⚖ Ganesh's Pattern	
	💡 Hollow Diamond Pattern(Pattern 6)	
	💡 Hollow Rhombus Pattern	
	💡 Pascal Triangle 1	
	💡 Pattern Numbers & Stars - 1	
	💡 Pattern Numbers & Stars - 2	
-	Pointers and Functions (Concepts)	14/14 Lectures Completed
	CPP - Address Of Operator '&'	
	CPP - Pointers	
	CPP - Dereference Operator	
	CPP - Functions	
	CPP - Call Stack	
	CPP - Factorial and NCR Function	
	CPP - Pass by Reference Using Pointers	
	Stanford CS -Pointers and Memory	
	CPP Notes - Pointers	
	CPP - Pointers PDF	
	CPP Notes - Functions	
	MCQ - Pointers	
	MCQ - Pointers Solution	
	CPP Webinar : Pointers [17-Feb-2018]	
-	Arrays 1.0 (Problem Solving on 1D Arrays)	10/10 Lectures Completed

- ▶ CPP - Arrays introduction 
- ▶ CPP - Linear Search 
- ▶ CPP - Generating Subarrays 
- ▶ CPP - Maximum Subarray Sum 1 
- ▶ CPP - Maximum Subarray Sum 2 
- ▶ CPP - Maximum Subarray Sum 3 (Kadane's Algo) 
- ▶ CPP - Chewbacca and Number, HackerBlocks Tutorial 
- ▶ CPP - Help Ramu, Bruteforce, Hackerblocks 
- ▶ CPP - Arrays - Rainwater Trapping 
- ⬇ Editorial - Maximum Circular Subarray Sum 

- **Arrays 2.0 (2D Arrays, Strings)** 15/15 Lectures Completed

- ▶ CPP - Character Arrays | Using `cin.getline()` 
- ▶ CPP - Strings STL 
- ▶ CPP Webinar : 2-D Arrays, Strings 
- ▶ CPP - Staircase Search 
- ▶ CPP - STL String Tokenizer Function 1 
- ▶ CPP - STL String Tokenizer 2 
- ▶ Interview Problem - Image Rotation 
- ⬇ CPP Notes - Multidimensional Arrays 
- ▶ CPP - Vectors STL 
- ⬇ CPP - Vectors and It's Functions 
- ▶ Dynamic Memory Allocation In C++ Introduction 
- ▶ Concept of Heap In Dynamic Memory Allocation In C++ 
- ▶ Use Of `new` and `delete` Operator in C++ 

Allocating 2d Array Using
Dynamic Memory Allocation In
C++



Dynamic Memory Allocation In
C++ Conclusion



- Challenges - Arrays

1/17 Lectures
Completed

</> Arrays-Max Value In Array



</> Arrays-Wave print Column
Wise



</> Arrays-Target Sum Pairs



</> Arrays-Target Sum Triplets



</> Rain Water Harvesting



</> Maximum Subarray Sum



</> Maximum Circular Sum



</> Maximum length Bitonic
Subarray



</> Arrays-Spiral Print
Anticlockwise



</> Rotate Image (N X N Array)



</> Form Biggest Number



</> Chewbacca and Number



</> Prime Visits



</> Broken Calculator



</> Matrix Search



</> Arrays-Sum Of Two Arrays



</> Median of Sorted Arrays



- Challenges - Strings

2/12 Lectures
Completed

</> Lower Upper



</> Character Type



</> Strings-Max Frequency



Character

- </> 🌟 CanYouReadThis?
- </> 🌟 Is Palindrome?
- </> 🌟 Strings-Difference in Ascii Codes
- </> 🌟 SanketAndStrings
- </> 🌟 Piyush and Magical Park
- </> 🌟 String Compression
- </> 🌟 Ultra-Fast-Mathematicians
- </> 🌟 Finding CB Numbers
- </> 🌟 Brackets All Over

- Sorting & Searching (Divide and Conquer) 15/15 Lectures Completed

- ▶ CPP - Bubble Sort
- ▶ CPP - Selection Sort
- ▶ CPP - Insertion Sort
- ▶ CPP - Wave Sort
- ▶ CPP - Counting Sort
- ▶ CPP - Inbuilt Sort Function
- ▶ Comparators
- ▶ CPP - Sanjay's New Job, Sorting, Hackerblocks
- ▶ CPP - Binary Search Algorithm
- ▶ CPP - Binary Search-2 Implementation & Complexity
- ▶ Binary Search: Upper & Lower Bound
- ▶ Binary Search : Square Root
- ▶ Binary Search: Searching in Sorted Rotated Array
- ▶ Binary Search: Pivot Element
- ▶ Binary Search - Book Allocation

Binary Search . Book Allocation Problem		
-	Challenges - Sorting & Searching	11/18 Lectures Completed
▶	Quiz On Sorting	✓
▶	Quiz On Divide And Conquer	✓
↔	💡 Arrays-Bubble Sort	🔴
↔	💡 Arrays-Selection Sort	✓
↔	💡 Arrays-Insertion Sort	✓
↔	💡 SORTING IN LINEAR TIME	✓
↔	💡 String Sort	✓
↔	💡 Sort Game	✓
↔	💡 Counting Sort	✓
↔	💡 Grand Temple	🔴
↔	💡 Arrays-Binary Search	✓
↔	💡 KTH ROOT	✓
↔	💡 Find Upper and Lower bound	✓
↔	💡 Book Allocation Problem	🔴
↔	💡 Aggressive Cows	🔴
↔	💡 Winning CB Scholarship	🔴
↔	💡 Painter Problem	🔴
↔	💡 HELP RAHUL TO SEARCH	🔴
-	Bitmasking Techniques	6/6 Lectures Completed
▶	CPP - Bitwise Operators	✓
⬇	CPP Code - Finding Set Bits	✓
▶	Bitmasking Basics(Webinar)	✓
▶	Unique Number-III, Hackerblocks, Bitmasking	✓

▶ CPP - Incredible Hulk Problem	✓
▶ Tavas and SaDDas, Codeforces 535-B	✓
- Challenges - Bitmasking	1/9 Lectures Completed
◀/▶💡 Playing With Bits	✓
◀/▶💡 Unique Number - I	🔴
◀/▶💡 XOR Profit Problem	🔴
◀/▶💡 Count Set Bits	⚪
◀/▶💡 Unique Numbers - II	⚪
◀/▶💡 Unique Number - III	⚪
◀/▶💡 Incredible Hulk	⚪
◀/▶💡 Count Subsequences	⚪
◀/▶💡 Power(O(logn))	⚪
- Number Theory Basics	8/8 Lectures Completed
▶ Checking Primes - Trial & Division Method	✓
▶ Sieve of Eratostenes	✓
▶ Optimisations in Prime Sieve	✓
▶ GCD 01 - Euclid's Algorithm Concept	✓
▶ GCD 02 - LCM, GCM Code	✓
▶ Pigeonhole Principle	✓
▶ CPP - Divisible Subarrays, Pigeonhole Problem	✓
▶ Modulo Properties	✓
- Challenges - Number Theory	0/6 Lectures Completed
◀/▶💡 GCD	⚪
◀/▶💡 LCM	⚪

 Deepak and Primes 

 DIVISIBLE SUBARRAYS 

 Modular Exponentiation 

 Editorial - Deepak and Prime [Sieve] 

- Recursion and Backtracking 3/23 Lectures Completed

 CPP - Understanding Recursion 

 CPP - Fibonacci Number Recursion 

 CPP - Recursion Array is Sorted 

 CPP - Time to Try - Power Fn 

 CPP - Fast Power 

 CPP - Recursion Increasing Decreasing Numbers 

 CPP - Recursion - Time to Try 

 CPP - Recursion - Tiling Problem. 

 Recursion - 2048 Problem 

 CPP - Recursion - Linear Search 

 CPP - Recursion Time To Try - Binary Search 

 CPP - Recursion Bubble Sort 

 Tower of Hanoi 

 CPP - Merge Sort 

 QuickSort 

 Randomized Quicksort 

 CPP - Recursion - Subsequences 

 CPP - Recursion - Phone Keypad 

 CPP - Permutations 

 N-Queen Using Backtracking 

▶ CPP - Recursion Rat in a Maze



▶ N-Queen using Bitsets



▶ CPP - Recursion - Sudoku Solver



- ⚡ Challenges - Recursion 0/21 Lectures Completed

◀/▶💡 Sorted Array



◀/▶💡 R9 -- Found At Last



◀/▶💡 Recursion-Convert String to Integer



◀/▶💡 Recursion-Duplicate character formatting



◀/▶💡 Strings-Remove Duplicates



◀/▶💡 All Indices Problem



◀/▶💡 Tiling Problem - II



◀/▶💡 Tower of Hanoi



◀/▶💡 Replace all π



◀/▶💡 Recursion-Subsequences



◀/▶💡 Mapped Strings



◀/▶💡 Move All X at End



◀/▶💡 MERGESORT



◀/▶💡 QUICKSORT



◀/▶💡 Subset Sum Easy



◀/▶💡 Smart Keypad - I



◀/▶💡 Smart Keypad - Advanced



◀/▶💡 Recursion-Keypad Codes



◀/▶💡 Optimal Game Strategy-I



◀/▶💡 Recursion-Dictionary Order(Larger)



◀/▶💡 Recursion-Dictionary Order(smaller)



- Challenges - Backtracking

0/8 Lectures
Completed

- Class Assignment
- SUDOKU SOLVER
- COUNT N-QUEEN
- Rat in a maze
- Sum It Up
- Tricky Permutations
- Generate Parentheses
- Funky Chess Board

- Space Time Complexity Analytics

0/10 Lectures
Completed

- CPP Webinar : Space Time Complexity [28 Feb]
- Time Limit Exceeded - Learn about Complexities !
- CPP Notes - Dynamic Memory | Space Time Complexity
- CPP Code - Space Time Complexity Webinar
- CPP Notes - Time Complexity
- CPP Notes - Dynamic Memory Allocation
- CPP Notes - References in C++
- MCQ - Complexity Analysis
- MCQ - Complexity Analysis Solution
- Quiz on Time and Space Complexity

- Object Oriented

1/19 Lectures

Programming Concepts

Completed

- ▶ CPP OOPS 1 - Introduction to Object Oriented Programming 
- ▶ CPP OOPS 2 - Classes, Data Members and Functions 
- ▶ CPP OOPS 3 - Getters and Setters 
- ▶ CPP OOPS 4 - Constructor and Parameterised Constructor 
- ▶ CPP OOPS 5 - Copy Constructor 
- ▶ CPP OOPS 6 - Shallow and Deep Copy 
- ▶ CPP OOPS 7 - Copy Assignment Operator 
- ▶ CPP OOPS 8 - Destructors 
- ▶ CPP OOPS 9 - Initialization List, Consts 
- ⬇ CPP OOPS Codes 
- ▶ CPP Webinar - OOPS Webinar | Operator Overloading 
- ▶ OOPS Webinar II - Vectors and C++ STL 
- ⬇ CPP Notes - OOPs I : Class, Objects, Access Modifiers, Friend Class, Getters and Setters 
- ⬇ CPP Notes - OOPs III : Function Overloading and Operator Overloading 
- ⬇ CPP Notes - OOPS II : Constructors, Destructors, Const Data Members 
- ⬇ CPP Notes - OOPs IV 
- ⬇ MCQ - Object Oriented Programming 
- ⬇ MCQ - Object Oriented Programming Solution 
- ▶ Quiz On Object Oriented Programming In C++ 

- Linked Lists

0/16 Lectures Completed

- ▶ CPP - Data Structures Introduction 
- ▶ CPP - Linked List Introduction 

▶ CPP Linked List - Insert at Head



▶ CPP- Linked List Insertion in Middle



▶ CPP - Linked List Deletion



▶ CPP - Linked List Searching



▶ CPP Linked List - Taking Input



▶ CPP Linked List - Operator Overloading



▶ CPP - Linked List Reverse



▶ CPP - Recursive Reverse a Linked List



▶ CPP Linked List - Midpoint | Runner Technique



▶ CPP Linked List - Kth Node from the End



▶ CPP Linked List - Merge two sorted Linked Lists



▶ CPP Linked List - Merge Sort



▶ CPP - Linked List Floyd's Cycle



⬇ CPP Notes on Linked List



- ⚡ Challenges - Linked Lists

0/11 Lectures Completed

▶ Quiz On Linked List



</>💡 LL - k Reverse



</>💡 Merge Sorted Linked Lists



</>💡 Linked List-K Append



</>💡 Even-After-Odd



</>💡 Kth element from last in linked list



</>💡 Intersection point two linked lists



</>💡 Circular Linked List



</>💡 Palindrome



 Rat Chases its cheese



 Sort linked list using insertion sort



- Stacks and Queues 0/13 Lectures Completed

 CPP Data Structures - Stack Introduction



 CPP - Stack Implementation



 CPP - Stack Implementation using Templates



 CPP - Stack STL



 CPP Data Structure - Queue Introduction



 CPP - Queue Implementation



 CPP - Queue STL



 CPP - Reverse Stack Using 1 Extra Stack



 CPP - Stack Reverse Using Recursion



 CPP - Stack using 2 Queues



 CPP - Stacks Question : Balanced Parenthesis



 CPP - Stock Span Problem



 CPP - Stack - Histogram Area



- Doubly Ended Queue & Sliding Window 0/3 Lectures Completed

 CPP - Doubly Ended Queue



 Interview Problem - Maximum element in window of size k



 Interview Problem - Maximum Length Unique Character Substring | Sliding Window



- ⚡ Challenges - Stacks,Queue,Deque 0/12 Lectures Completed

 Quiz On Stacks And Queues



 🔍 Find the greater element



- Prateek Sir and Coding
- Playing with cards (In stack)
- Balanced Parenthesis
- Stock Span
- HISTOGRAM
- Redundant Parentheses
- Dequeue efficient queue using stack
- REVERSE A STACK USING RECURSION
- Strongest Fighter
- Importance of Time

- Binary Trees & Binary Search Trees 0/22 Lectures Completed

- CPP - Binary Tree - Introduction
- CPP - Binary Tree - Preorder Build and Print
- CPP - Binary Tree - Inorder and Postorder Traversal
- CPP - Binary Tree - Level Order Print Recursive
- CPP - Binary Tree BFS Traversal-I
- CPP - Binary Tree Level Order Traversal - II
- CPP - Binary Tree - Count and Sum Nodes
- CPP - Binary Tree - Diameter of Tree
- CPP - Binary Tree - Diameter of the Tree Optimized Approach
- CPP - Binary Tree Question - Sum Replacement
- CPP - Binary Tree Solution - Sum Replacement
- CPP - Binary Tree - Height Balanced Tree
- CPP - Binary Tree - Build Balanced Tree From Array

▶ CPP - Binary Tree - Build Tree from PreOrder and Postorder



▶ CPP - Binary Search Tree - Introduction



▶ CPP - Binary Search Tree - Insertion & Build



▶ CPP - BST - Searching



▶ CPP - BST - Deletion



▶ CPP - BST - Check for BST



▶ CPP - BST to Sorted Linked List
Convert / Flatten a Tree



▶ CPP - BST - Construct from Preorder



▶ Mathematics - Catalan Numbers/Counting BST's



- ⚡ Challenges - Trees

0/16 Lectures Completed

▶ Quiz On Trees



💡 Structurally Identical (Binary Tree)



💡 Build BST



💡 Trees -- Find sum at level K



💡 Print BST keys in the given range



💡 Tree bottom view



💡 Is_Balanced (Binary Tree)



💡 Level Order (Zigzag)



💡 Create tree (Using preorder and inorder)



💡 Replace with Sum of greater nodes



💡 Delete nodes from BST



💡 Tree top view



💡 Tree Left View



💡 Tree Right View



  Largest BST In A Bt



  Print all nodes at distance k from a given node



- Heaps 0/11 Lectures Completed

 CPP - Heaps - Introduction



 CPP - Heap Build & Insert



 CPP - Heap Deletion and Heapify Concept



 CPP - Heap Class Implementation I



 CPP - Heap Class Implementation II



 Priority Queue - STL I



 Priority Queue STL - II Functors



 Heaps Interview Q - Largest K elements in Running Stream - I



 Heaps Interview Q - Largest K Elements II - STL Implementation



 Heaps - Merge K Sorted Arrays



 Heaps - Running Median (Spoj - RMID)



-  Challenges - Heaps 0/4 Lectures Completed

 Quiz On Heaps



  Hostel Visit



  Merge K sorted Arrays



  Top k most frequent number in a stream



- Hashing 0/3 Lectures Completed

 LP Live Advanced - Hashing



 LP Live Advanced - Hashing(contd.)



 Hashing C++ STL



- Tries 0/1 Lectures Completed

▶ Trie - Data Structure (Concept + Implementation) 

- ⚡ Challenges - Hashing & Tries 0/15 Lectures Completed

▶ Quiz On Hashing 

◀/▶💡 Unlock 

◀/▶💡 Exist OR Not 

◀/▶💡 Arrays-Intersection Of Two Arrays 

◀/▶💡 Vertical Order Print Binary Tree 

◀/▶💡 String Window 

◀/▶💡 Highest frequency(Hashing) 

◀/▶💡 Mike and HashTrick 

◀/▶💡 Subarrays with distinct elements 

◀/▶💡 Digital Dictionary 

◀/▶💡 Maximum XOR Subarray 

◀/▶💡 Range Xor 

◀/▶💡 Trie Interview 

⬇️ Editorial - Trie Interview 

⬇️ Editorial - Range Xor 

- Graph Algorithms 0/19 Lectures Completed

▶ Graphs 01- Introduction 

▶ Graphs 02 : Representation (YT) 

▶ Graphs 03- Adjacency List Implementation for Integer Data 

▶ Graphs 04 - Adjacency List Implementation for Generic Data 

▶ Graph 05- Breadth First Search



▶ Graphs 06: BFS - Single Source Shortest Path



▶ Graphs 07- Snakes and Ladder Problem using BFS-SSSP Algorithm



▶ Graphs 08- Depth First Search



▶ Graph 09- Connected Components using DFS



▶ Graph 10- Topological Sort Using DFS



▶ Graph 11- Topological Sort Using BFS



▶ Graph 12- Cycle Detection in Undirected Graph using BFS



▶ Graph 13- Cycle Detection in Directed Graph using DFS



▶ Dijkstra's Algorithm Shortest Path on Weighted Graphs



▶ Bellman Ford Algorithm - Graphs with Negative Edges



▶ ICPC Trip - Interesting Graphs Problem!



▶ DP with Bitmasks : Travelling Salesman Problem (2-d DP)



▶ Holi- Spoj, Pigeonhole + Graphs



▶ Advanced Graphs Webinar - Bipartite Graph, Strongly Connected Components, Minimum Spanning Tree, Union-Find, Problem Solving



- Challenges - Graphs 0/7 Lectures Completed

▶ Quiz On Graphs



</>💡 Dijkstra's Algorithm



</>💡 Religious People



</>💡 Pairing



</>💡 BFS - Shortest Path



</>💡 Snake and Ladders Board (BFS)



</>💡 Beautiful Vertices



- Dynamic Programming 1/16 Lectures Completed

- ▶ CPP ★ Dynamic Programming | How to get maximum from this playlist
- ▶ CPP ★ Dynamic Programming | Need for DP and Memoization
- ▶ CPP ★ Dynamic Programming | Bottom up approach
- ▶ CPP ★ Dynamic Programming | Reduce a number to one in minimum number of step (1-D DP)
- ▶ CPP ★ Dynamic Programming | Wine Problem (2-D DP)
- ▶ CPP ★ Dynamic Programming | When to use DP, Grid Problem
- ▶ CPP ★ Dynamic Programming | Rod cutting to maximize profit (1-D DP)
- ▶ CPP ★ Dynamic Programming | Edit Distance
- ▶ CPP ★ Dynamic Programming | Longest Increasing Subsequence
- ▶ CPP ★ Dynamic Programming | Longest Common Subsequence
- ▶ Advanced DP : k-Ordered LCS, HackerEarth
- ▶ Advanced DP : Cell Mitosis, HackerBlocks
- ▶ Advanced DP : Robot Paths, Codechef
- ▶ Advanced DP : Mixtures, Spoj
- ▶ CPP Dynamic Programming - Friends Pairing Problem
- ▶ Dynamic Programming - Optimal Game Strategy

- ⚡ Challenges - Dynamic Programming 0/17 Lectures Completed

- ▶ Quiz On Dynamic Programming
- ◀ Print LCS
- ◀ Count Number of Binary Strings

- 0-1 Knapsack
- 0-N Knapsack
- LCS with 3 Strings
- K-Ordered LCS
- 1 Count
- Minimum Money Needed
- Exchanging Coins
- Wildcard Pattern Matching
- The Subset Sum To Target
- Minimum Jumps Required
- Mixtures
- Ugly Numbers
- Optimal Game Strategy-I
- Optimal Game Strategy-II

- Greedy Algorithms 0/5 Lectures Completed

- Live Class 06 - Greedy Algorithms
- Maximum Circles
- Deepak and his journey
- Mike and HashTrick
- Quiz On Greedy Algorithm

- String Matching Algorithms 0/2 Lectures Completed
Due 11-Jan-19

- Rabin-Karp And Naive String Matching Algorithm - Concept
- Rabin-Karp String Matching Algorithm - Code

- Interview Puzzles 4/4 Lectures Completed

- ▶ CPP - Apples and Oranges 
- ▶ Puzzle Hour Glass 
- ▶ Puzzle - Infinite quaters 
- ▶ Interview Puzzle - Celebrity Problem 