



Code
Greedy

How to Approach a DSA Problem & Develop Logic



What will we discuss

How Top Problem Solvers Think

How To Develop Logic

Code Quality

How to Approach a Problem

Perfect Order to Learn Data Structure

Which Language to choose.

Pre-requisite for DSA

Platform to practice

Your Doubts....

Leetcode

 **Pramod Bhosale**
Software Engineer
0 Followers • 1 Following

Edit Profile  

Overview Coding Score Posts Bookmarks

Coding Score **1512** Problems Solved **530**

Institute Rank **3** Articles Published **—**

 **0 Day POTD Streak**

Longest Streak: **56 Days** POTDs Solved: **175**

Contest Rating **1,641** Global Ranking **143,195/802,558**
Rank **120,461**

Figuring Out 

 **Pramod Bhosale** 
PramodBhosale67

TCS DIGITAL | Software Engineer

India

shivaji University, Kolhapur

Pramod-Bhosale67

pramodbhosale-

c++ database operating-system data-structures

Community Stats

Views 7.5K Last week +17 Solution 17

Made with Goodnotes

339 Submissions in Year [2025](#) ▾

Year Month

586 / 3778 ✓ Solved 53 Attempting

Easy 213/917 Med. 316/1969 Hard 57/892

Badges 8 →

Most Recent Badge **500 Days Badge**

977 submissions in 2023

Total active days: 250 Max streak: 28 2023

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

GFG

Why DSA

Problem solving skills.

Highest Package -----> DSA vs Dev

Scalability -----> Wide Applications



Placement ke sath bhi
Placement ke baad bhi

Startup → less DSA | more development

MNC (TCS, infys, wipro
capg) → fundamentals (2)

DSA) SD

Sept 2022 → 7.5 I,

(3)

Product Based

Dce → 11.5

(Banc L → 27)

Barclay

Start career

2nd → DSA → 3rd 2nd SCM

3rd → M-F DSA ←

S-S → WC

How Weak Problem Solvers Think

How many of you read a problem and directly jump to code?"

30 min

How many of you feel: 'I understand the solution... but I can't think of it myself'?



Which Language to choose.



Pre-requisite for Language

CPP: Standard Template Library

JAVA: Collections Framework

Python: Libraries



Perfect Order to Learn Data Structure

- ~~Array~~
- ~~String~~
- ~~2D Array~~
- ~~Searching Sorting~~
- ~~Recursion~~
- ~~Linked List~~
- ~~Stack Queue~~
- ~~Bit Manipulation~~
- ~~Greedy Algorithms~~
- ~~Tree~~
- ~~Graph~~
- ~~Dynamic Programming~~
- ~~Tries~~
- ~~String Hard level~~
- ~~Recursion Hard Level~~



Typing Speed Matters

Typing Master

keybr.com

Typing master





HOW TO DEVELOP LOGIC



Start with Easier problems



Problem to learn, Not to solve

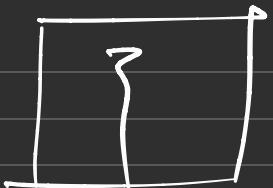
Pen & Paper are your best friend

3D

When to see the solution

Prove your logic, before submitting

Consistency



$$X = 7, Y = 8 \\ N = 10$$



WAY TO LEARN ANY TOPIC



- 01 Chose a topic which you're learning
- 02 Learn all it's theory & applications
- 03 Implement the DS & it's applications
- 04 Pick 20 Questions 30
- 05 10 Easy 10 Medium 15 5 Hard

From where to learn Topics or find problems

CodeGreedy's Pro DSA Sheet

Welcome to CodeGreedy's Pro DSA Sheet – Loved by Students, Trusted by Toppers

This isn't just another collection of problems—it's a battle-tested roadmap designed to turn you into a real problem solver, not just a copy-paste coder.

What Makes It Different?

- ↗ Zero to Hero Journey: Whether you're a beginner or brushing up for MAANG, this sheet takes you step-by-step from fundamentals to advanced topics like DP, Graphs
- 👉 Handpicked Problems, CodeGreedy Style: Each topic is backed by carefully chosen LeetCode problems, simplified and explained with our signature easy-to-understand approach.
- 👤 Structured for Placements: Patterns, strategies, and mindsets—all tailored for cracking real interviews, not just clearing topics.
- ❤️ Built for the Community: Thousands of learners have already trusted this sheet—not because of the name, but because it works.

How much time to give for a single Problem

5 min - Read the Problem

5 min - Understand the
Input Output

30 min - Think Different
solutions

Watch HINTs

Again think for 10 min

If not, watch Solution



Code Quality

HOW TO APPROACH THE PROBLEM

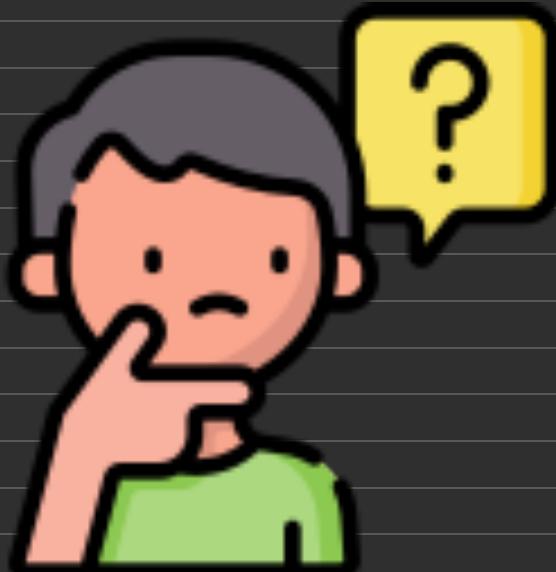


- 01 Read the Problem Statement.**
- 02 Test cases & get the obeservation.**
- 03 Think! How the results are obtained.**
- 04 Now, think, which DSA can be used.**
- 05 Brute -> Better -> Optimal**

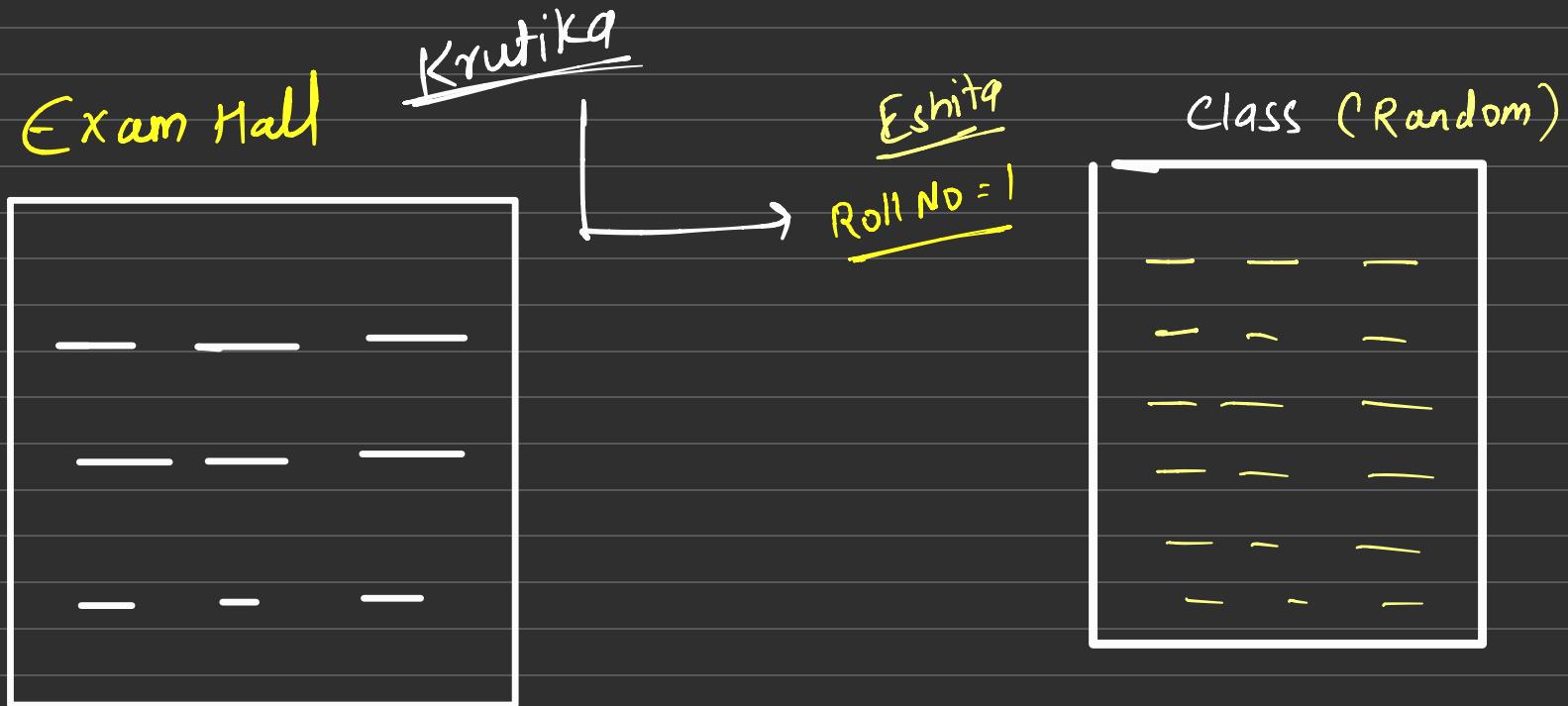
Platform to practice



Your
Doubts



What is Data Structure And why do we need it.



Type of Data structure

Linear

Array, Linked List, Stack, Queue

Non - Linear

TREE, Graph

Hashing

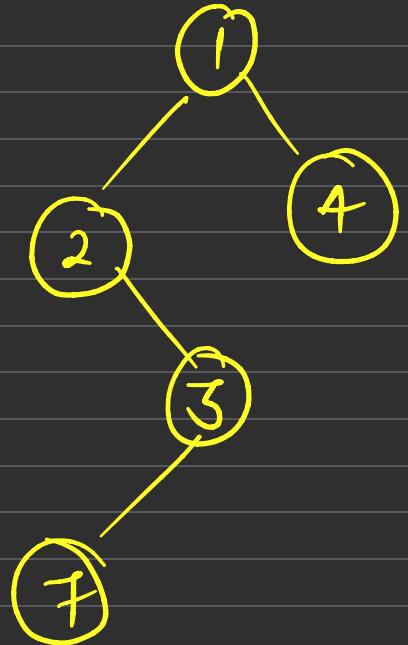
Linear:

- Sequentially ordered / linear Order



Non-linear

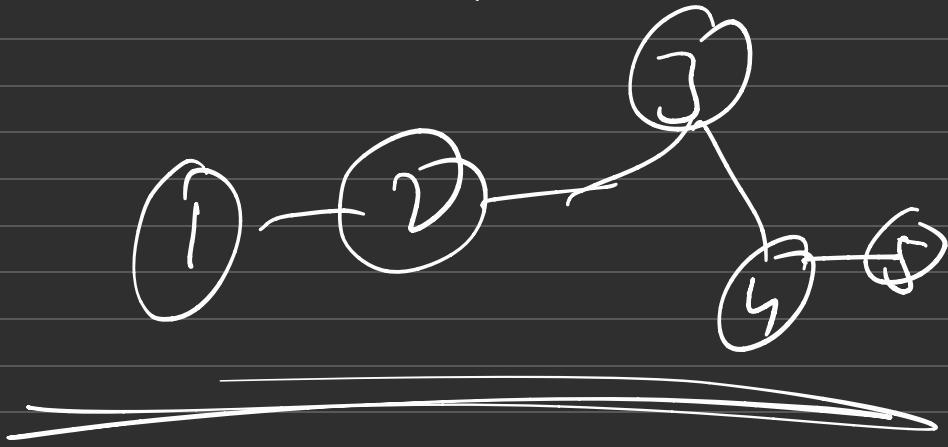
- Randomly
- Level by level
- Dynamic



NL

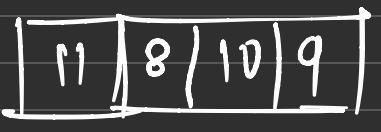
1	4	6	3
---	---	---	---

sin car



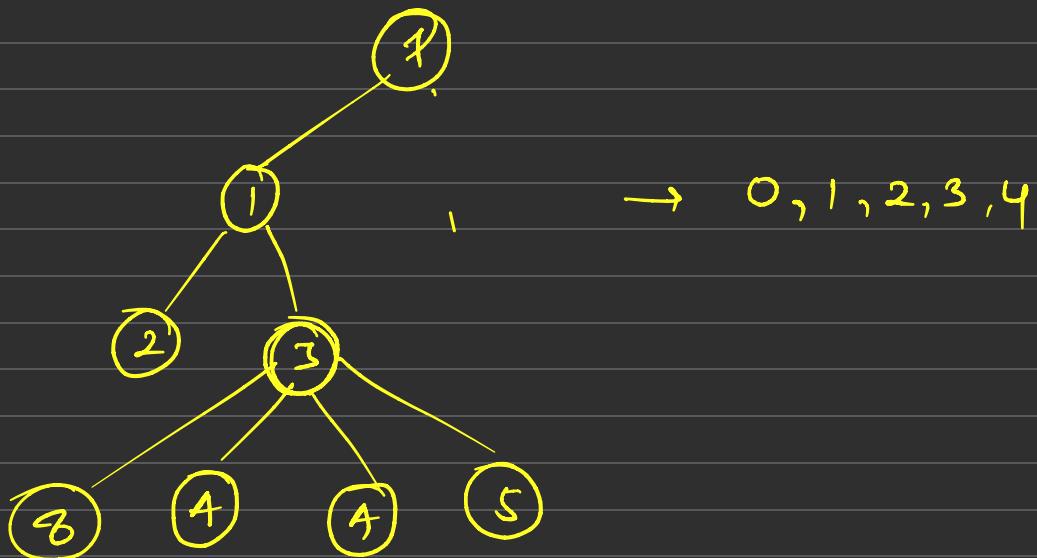
linear \rightarrow Every element will have atmost 2 neigh

{ 0, 1, 2 }

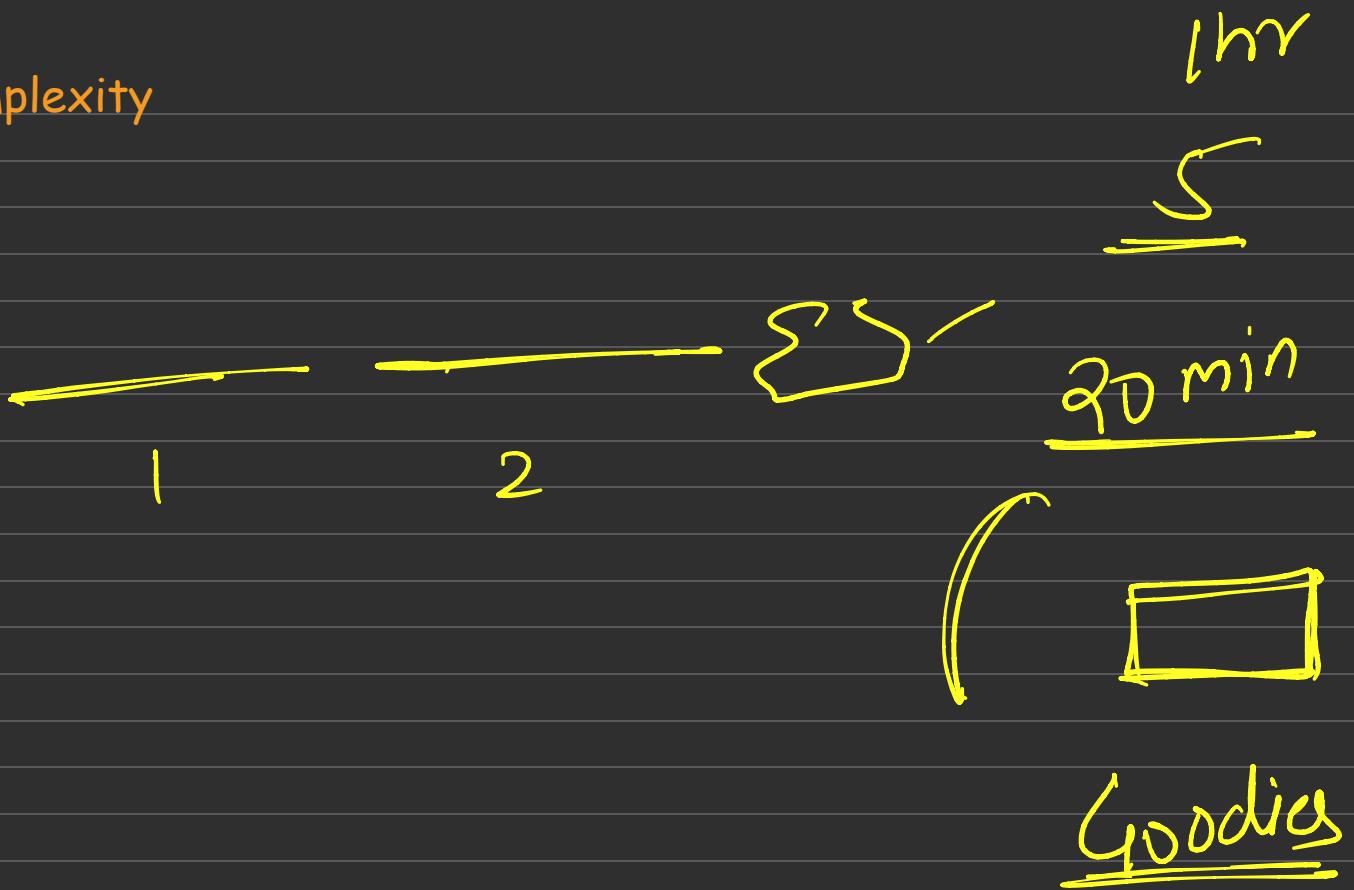
arr =  \rightarrow 0, 1, 2

Non-linear:

Every element will have at least 2
neigh.



Time Complexity



input size	required time complexity
$n \leq 10$	$O(n!)$
$n \leq 20$	$O(2^n)$
$n \leq 500$	$O(n^3)$
$n \leq 5000$	$O(n^2)$
$n \leq 10^6$	$O(n \log n)$ or $O(n)$
n is large	$O(1)$ or $O(\log n)$

S.NO	Big O Notation	Name
1.	$O(1)$	Constant Time Complexity
2.	$O(\log n)$	Logarithmic Time Complexity
3.	$O(n)$	Linear Time complexity
4.	$O(n \log n)$	Linearithmic Time Complexity
5.	$O(n^2)$	Quadratic Time Complexity
6.	$O(n^3)$	Cubic Time Complexity
7.	$O(n^y)$	Polynomial Time Complexity
8.	$O(2^n)$	Exponential Time Complexity
9.	$O(n!)$	Factorial Time Complexity

Algorithm (applied to Array)	Time Complexity		
	Best Cases	Average Cases	Worst Cases
Bubble sort	$O(n)$	$O(n^2)$	$O(n^2)$
Selection sort	$O(n^2)$	$O(n^2)$	$O(n^2)$
Insertion sort	$O(n)$	$O(n^2)$	$O(n^2)$
Shell sort	$O(n \log(n))$	$O(n \log^2(n))$	$O(n \log^2(n))$
Merge sort	$O(n \log(n))$	$O(n \log(n))$	$O(n \log(n))$
Quick sort	$O(n \log(n))$	$O(n \log(n))$	$O(n^2)$
Heap sort	$O(n \log(n))$	$O(n \log(n))$	$O(n \log(n))$
Counting sort	$O(n+k)$	$O(n+k)$	$O(n+k)$
Bucket sort	$O(n+k)$	$O(n+k)$	$O(n^2)$
Radix sort	$O(nk)$	$O(nk)$	$O(nk)$