



TOP CODERS LAST MINUTE PREPARATION

by Manas Tiwari

ABOUT ME



Upcoming Microsoft SDE



Cracked Zomato On-Campus

Interviewed For



media.net

CODING PROFILES



CODEFORCES
EXPERT | RATING : 1830

ACHIEVEMENTS



META HACKER CUP
AIR #58 | World Rank #634

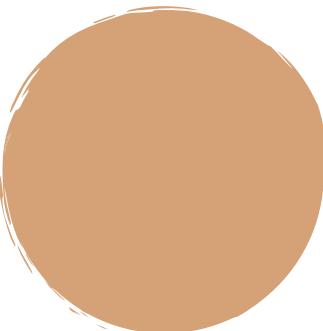


GOOGLE KICK START
AIR #175 | World Rank #332

MENTORED

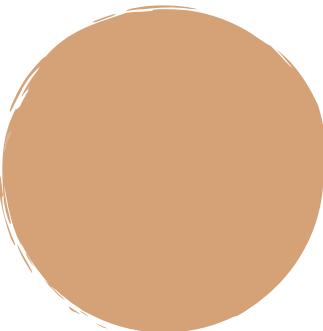
- 2020 Batch for Placement Training
- Juniors into CP as GDSC CP Lead
- Guided Training for coming placements

STRATEGY?



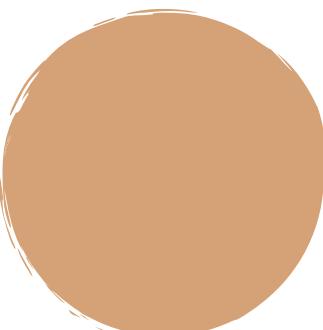
NO HUGE TEST CASES

Start by writing brute-force for every problem, if that can be written quickly.



SLOW JUDGE

Judge might take more than a minute to evaluate. Be accurate and proof-read everything before submitting



CODE WON'T BE CHECKED

It will not be possible to check everyone's code. So questions where a certain constraint is required, won't be present.



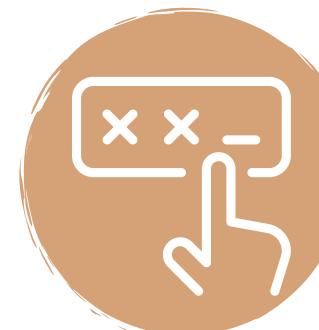
NO CUSTOM CHECKER

Judge can just match your output with a string output already present. So, no pointer manipulation questions. Solve everything as an array.



ATTEMPT EVERY QUESTION

You are likely to be scored on the number of test cases passed rather than the number of questions passed.



BE READY FOR WEIRD INPUT

In leetcode you are already given the input. In codeforces, you are given proper rules for input. Do not expect the same. In C++, know the stringstream class. In python, handling input is easier.

READ A LINE IN C++

To read a line in C++

- Use getline

To read a string as a stream of input

- Use stringstream

The following code can be used to read an array of elements from a single line.

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    string s;
    getline(cin,s);
    stringstream ss(s);

    vector<int> vec;
    int x;
    while ( ss >> x )
        vec.push_back(x);

}
```



PRACTICE PROBLEMS

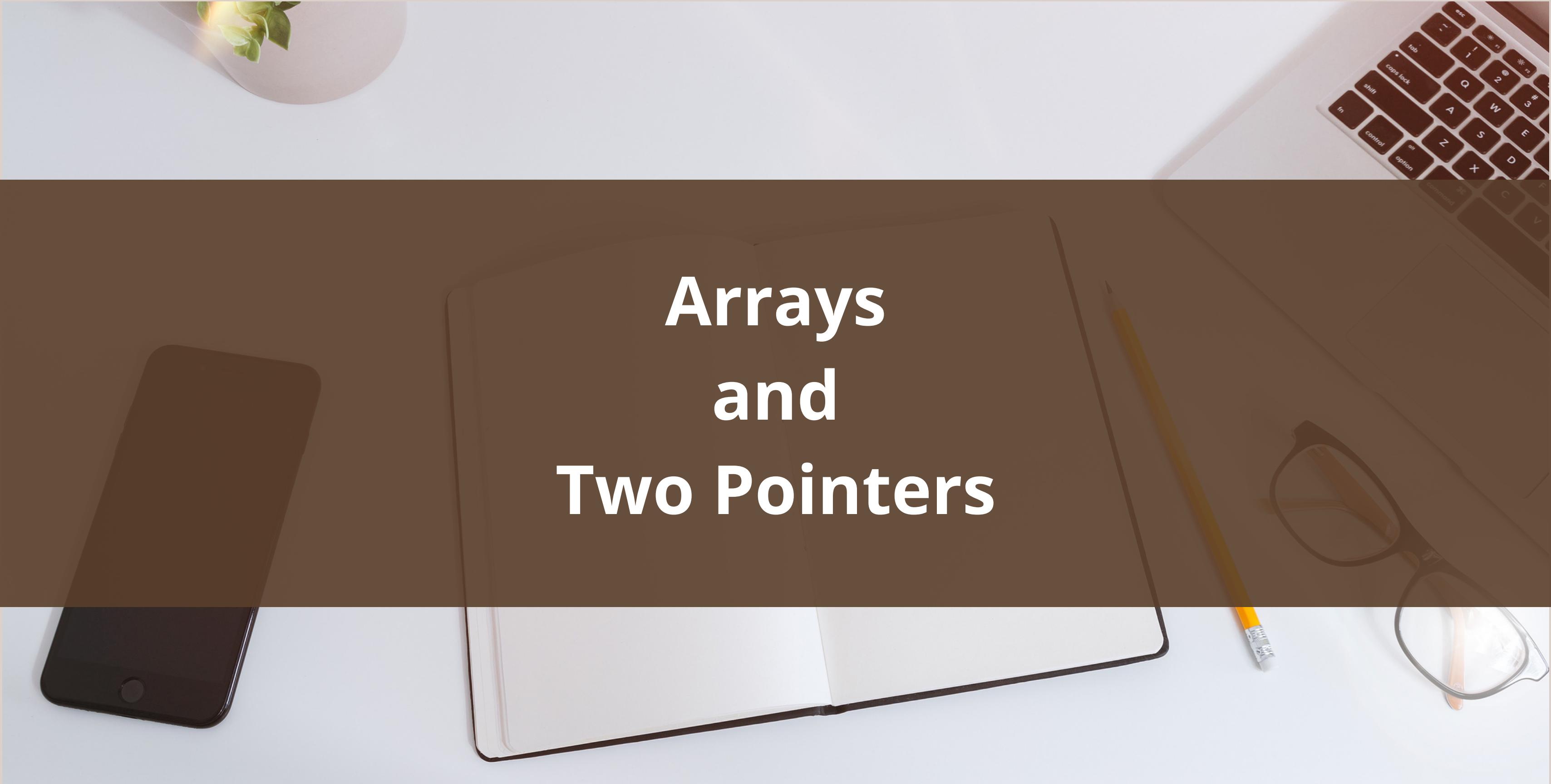
HOW ARE THE PROBLEMS PICKED?

The problems have been hand-picked by me based on the constraints discussed, and the topics that are likely to occur. I've scanned the internet for common interview problems consisting a variety of sheets which fit the constraints discussed. I've also included problems from my own course as well, which are also hand-picked.

WILL THESE PROBLEMS BE ENOUGH?

No one can ever be sure, as this has never happened earlier, at least not at this scale. Please indulge with these problems at your own risk, as nothing can be guaranteed. It is just a smart guess.

Arrays and Two Pointers



Easy - Moderate

Buy and Sell stock

Buy and Sell stock - II

Product of Array except Self

Gas Stations

3-SUM

Largest Substr with Uniq Characters

Koko eating bananas

N-Meetings in One Room

Merge Intervals

Hard

Candy

Trapping Rainwater

Stacks and Heaps



Easy - Moderate

Valid Parenthesis

NGE - II

Area of Histogram

Connect n-ropes

Job Sequencing Problem

Hard

Maximal Rectangle

Data Stream Median

Linked List Binary Trees Binary Search Trees



Easy - Moderate

Cycle Detection

Width of Tree

Depth of Tree Diameter of Tree

Same Tree

Validate Tree

Sum : Root to Leaf

Min. Abs. Diff

Hard

Merge K-sorted Lists

LRU

Vertical Order Traversal

Path Sum of Tree

Just use arrays for :

K Group-Reverse

Palindrome

Maths

```
function pow_mod(x, n, m):
    y := 1
    while n > 0:
        if n is odd:
            y := y · x mod m
        n := ⌊n/2⌋
        x := x · x mod m
    return y
```

$(a \bmod m) + (b \bmod m) \bmod m = a + b \bmod m$

$(a \bmod m) - (b \bmod m) \bmod m = a - b \bmod m$

$(a \bmod m) \cdot (b \bmod m) \bmod m = a \cdot b \bmod m$

Easy - Moderate

Sieve

Ugly Num 2

Unique Digits

K-th Symbol

Subarr Sum Equals K

Kadane | CF1

Bitwise AND Of Number Range

Pow(x,n)

Hard

Repeated Digits

XOR-Beauty

Line Sweep Algorithm

Codeforces

CF1

CF2

CF3

CF4

Dynamic Programming and Recursion

Easy - Moderate

[Integer to Roman](#)

[Jump Game](#)

[Unique Trees](#)

[Perfect Squares](#)

[Divisor Game](#)

[Coin Change](#) [Sum Partition](#) [Knapsack](#)

[House Robber 1](#) [House Robber 2](#)

[LCS](#) [LIS](#) [Edit Distance](#)

[Word Break](#) [Rod Cutting](#)

[Min Grid Sum](#) [Maximal Square](#)

Hard

[Count Anagrams](#)

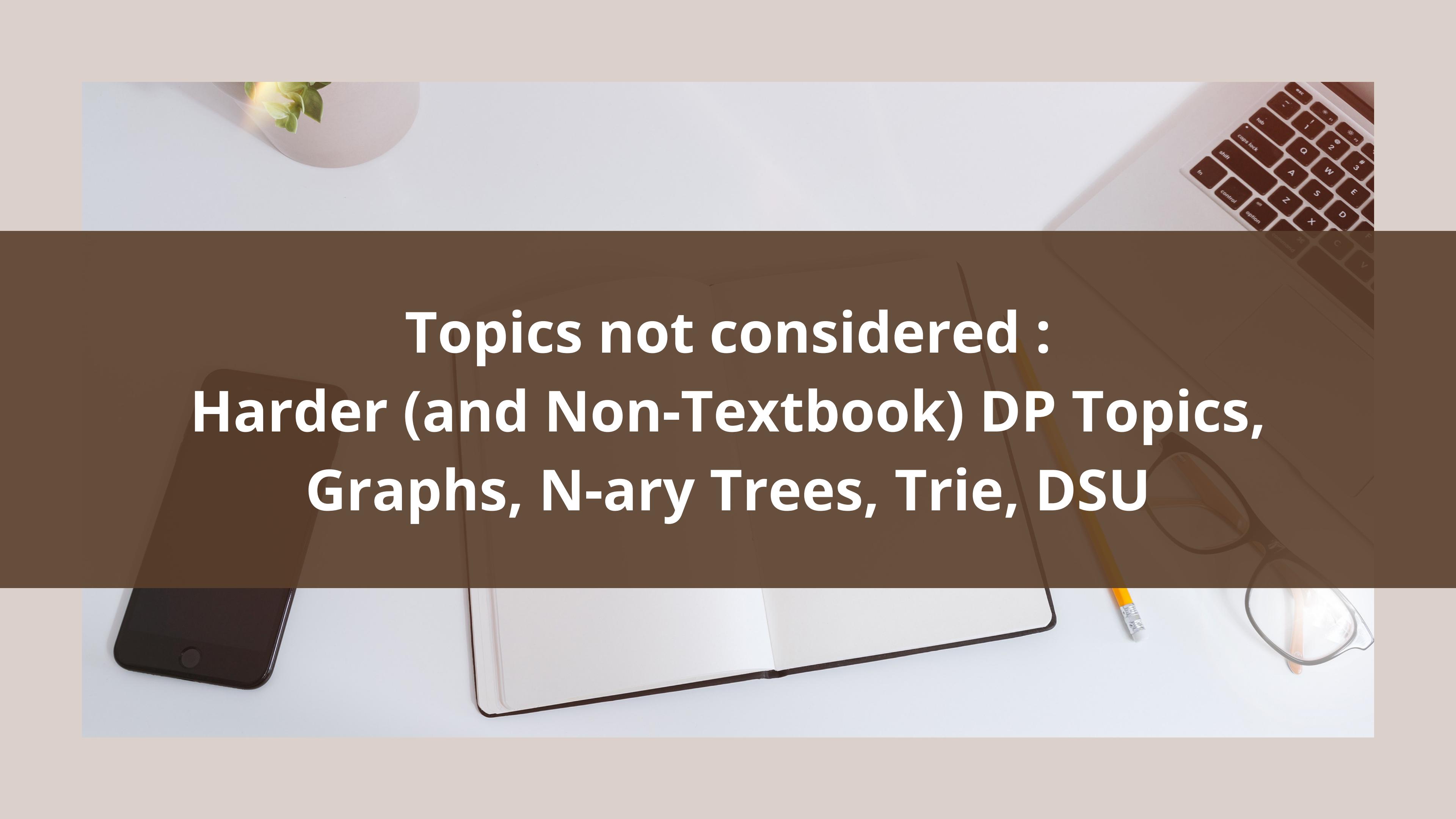
[Maximise Score](#)

[Word Search - II](#)

[N-Queens II](#)

[Stocks III](#) [Stocks IV](#)

[MCM](#)



Topics not considered :
Harder (and Non-Textbook) DP Topics,
Graphs, N-ary Trees, Trie, DSU

01

POSITIVE MIND

Whatever you have prepared is sufficient.
Perform well on whatever you know of.

02

RESET YOUR THINKING

Don't get stuck on a problem. Move on to a different one, and revisit it back again.
Works like a charm sometimes.

03

DON'T PANIC

This test is as new for everyone as it is for you. Calmly attend the questions.



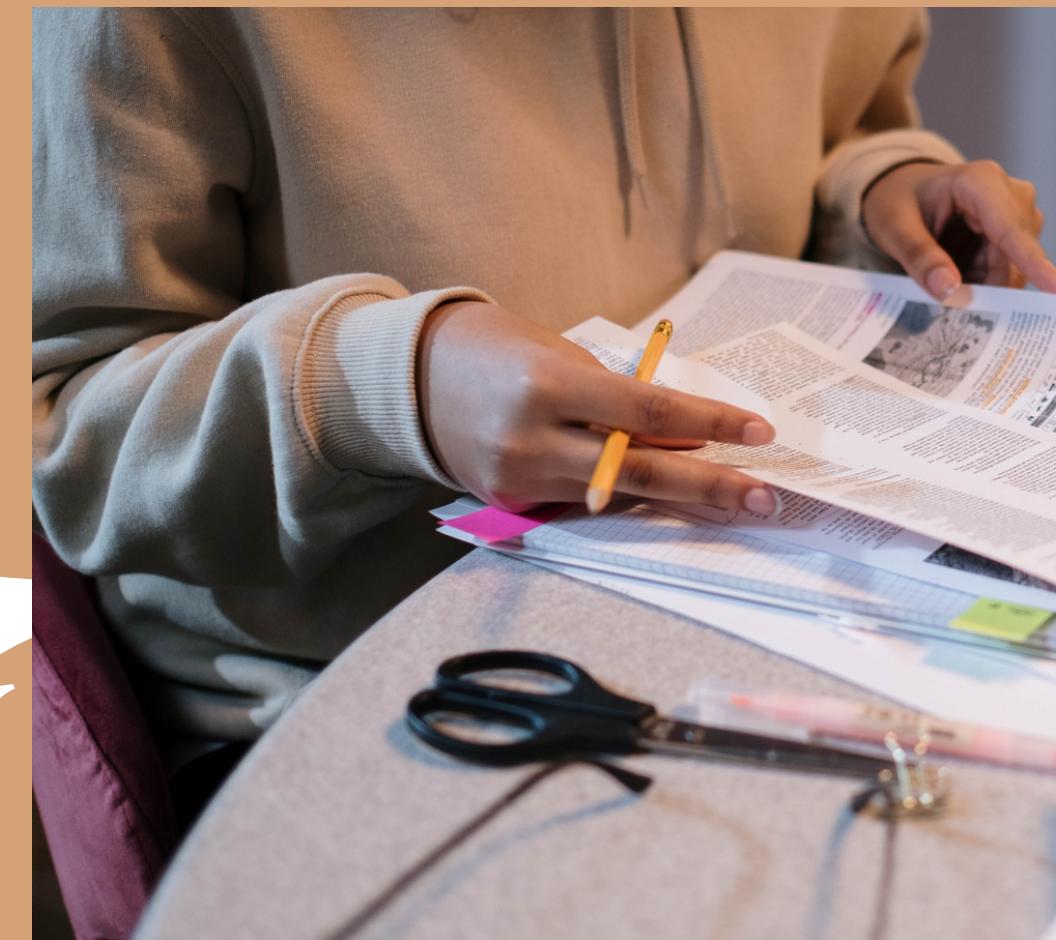
AFTERMATH?

I DID NOT PERFORM WELL IN THIS TEST. HOW SHOULD I COVER UP?

It is okay to mess up sometimes. Not everything goes as planned. Pull up your socks and prepare the harder topics like Harder DP, Graphs, N-ary trees, Trie, etc. You can still be shortlisted for good companies! In the end, this list will just help people get shortlisted and not placed. If you get shortlisted in companies, make sure you crack it till the very end. Have an astounding preparation when the placement drive starts.

CAN YOU RECOMMEND SOME RESOURCES TO PREPARE?

The resources on the internet are sufficient but are scattered everywhere. Sadly, the SDE Sheet alone isn't enough to crack the OAs and interviews. You require some blend of competitive programming type questions as well. Try visiting codeforces and search problems by tags. See if you can understand those problems side by side whatever you learn from sheets on the internet. Finding right problems will take time. You will require some mentoring and doubt support to speed it up.



GUIDED TRAINING



At this point most of you have learnt about basic to moderate algorithms and data structures, and may lack knowledge about DP variants, graphs, trees, and some special algorithms like DSU. I am guiding a few students around placement preparation for a small fee, and these are the next topics in our list.

HOW QUICKLY CAN I LEARN THESE?

That totally depends on your capability to absorb things, but you'll be doing it lot faster than what you can do alone.

OK, BUT WHAT DO YOU BRING TO THE TABLE?

The whole course is structured to be beginner friendly from very basic topics. There are two lectures and one doubt class every week with hand-picked problems from everywhere around the internet (both CP and DSA). I am there for doubt support 24x7. From next month we are starting the above mentioned harder topics from scratch. Join this group to be apart of it! [Link](#)





THANK YOU FOR YOUR ATTENTION

manastiwari111222



Manas Tiwari



_ma_na_s_



M-Manas-s

