9x9 Board → Existing values 1 to 9

- (1) Row 1 – 9 once
- (4) Column 1 – 9 once
- (4) 3x3 Matrix 1 – 9 once

mid/col
mid % col
log(m x n)

x[i][j]

1	2	9
3	4	7
6	8	5

arr[mid]

for (int i = 1; i <= 9; i++)
board[i][j] = i

Phone Keypad Problem

Letter Combinations of a given string: → remove

digits = "23"

2 → abc
3 → def

ad ae af
bd be bf
cd ce cf

3 → "def"

1 to 10
1 → " "
2 → "abc"

0 → " "

{1,2,3} { }
in
i en
{1,2,3} { }
{1,2,3} {1}

b
e
d

23
abc
ans
1 → def
ad ae af

Start: → letter Combination ("23")

Level 0: " " (empty, index = 0, digit = "2" → abc)

(3) (9) ans

→ 'a' → d → ad
→ e → ae
→ f → af
} backtrack

→ 'b' → d → bd
→ e → be
→ f → bf
} backtrack

→ 'c' → d → cd
→ e → ce
→ f → cf

bt

$3 \times 3 = 9$
0, 1, 2

Introduction to Greedy Algorithms:

All important & probable questions for interviews:

* Why greedy? Because find the max or min in some given data. (We save time)
→ Trick → (Try using Heaps)

* Minimum number of coins.
* Chocolate distribution problem.
or
Minimum Absolute Difference.
(Sliding Window Problem)
* Minimum cost of Connecting Ropes.
* Activity Selection Problem.
* Job Scheduling Problem.
* Policemen & Thieves.
* Huffman Encoding.
* Fractional Knapsack.
* Nikunj & Donuts.

* Minimum number of coins to reach a target value V

coins = {1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000}
(sort it)

int V = 91; (50, 20, 20, 1)
4 coins

V – 50 = 91 – 50
= 41 – 20
= 21 – 20
= 1 – 1 = 0

res = {50, 20, 20, 1}
res.size();
print(res) → coins

Chocolate Distribution Problem
Minimum Absolute Difference
Sliding window Problem:

arr = [7, 3, 2, 4, 9, 12, 56]   m = 3
↳ no of chocolates   students

(range) → m_x m_n (minimum)

sort = [2, 3, 4, 7, 9, 12, 56]

mini = Int Max

Math.min(mini, a)

mx		mn
4	–	2
7	–	3
9	–	4
12	–	7

2
4
5
5
56 – 9
Big