

AGENDA:

- Backtracking Intro
- Unique Subsets
- Generate Permutations
- Sudoku Solver

Unique Subsets

Given an array **arr[]** of integers of size **N** that might contain **duplicates**, the task is to find all possible unique subsets.

Note: Each subset should be sorted.

Example 1:

Input: N = 3, arr[] = {2,1,2}

Output: (), (1), (1 2), (1 2 2), (2), (2 2)

Explanation:

All possible subsets = (), (2), (1), (1,2), (2), (2,2), (2,1), (2,1,2)

After Sorting each subset = (), (2), (1), (1,2), (2), (2,2), (1,2), (1,2,2)

Unique Subsets in Lexicographical order = (), (1), (1,2), (1,2,2), (2), (2,2)

Example 2:

Input: N = 4, arr[] = {1,2,3,3}

Output: (), (1), (1 2), (1 2 3)

(1 2 3 3), (1 3), (1 3 3), (2), (2 3)

(2 3 3), (3), (3 3)

Generate Permutations

Given an array **arr[]** of unique elements. Generate all possible **permutations** of the elements in the array.

Note: You can return the permutations in any order, the driver code will print them in sorted order.

Examples:

Input: arr[] = [1, 2, 3]

Output: [[1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2], [3, 2, 1]]

Explanation: There are 6 possible permutations ($3! = 6$) of the array.

Input: arr[] = [1, 3]

Output: [[1, 3], [3, 1]]

Explanation: There are 2 possible permutations ($2! = 2$) of the array.

Sudoku Solver

Given an incomplete **Sudoku** in the form of matrix **mat[][]** of order 9*9, the task is to solve the Sudoku. It is **guaranteed** that the input Sudoku will have exactly **one** solution.

A sudoku solution must satisfy all of the following rules:

1. Each of the digits 1-9 must occur exactly once in each row.
2. Each of the digits 1-9 must occur exactly once in each column.
3. Each of the digits 1-9 must occur exactly once in each of the 9, 3x3 sub-boxes of the grid.

Note: Zeros represent blanks to be filled with numbers 1-9, while non-zero cells are **fixed** and cannot be changed.

Input: mat[][] =

3	0	6	5	7	8	4	0	0
5	2	0	0	0	0	0	0	0
0	8	7	0	0	0	0	3	1
0	0	3	0	1	0	0	8	0
9	0	0	8	6	3	0	0	5
0	5	0	0	9	0	6	0	0
1	3	0	0	0	0	2	5	0
0	0	0	0	0	0	0	7	4
0	0	5	2	8	6	3	0	0

Output:

3	1	6	5	7	8	4	9	2
5	2	9	1	3	4	7	6	8
4	8	7	6	2	9	5	3	1
2	6	3	4	1	5	9	8	7
9	7	4	8	6	3	1	2	5
8	5	1	7	9	2	6	4	3
1	3	8	9	4	7	2	5	6
6	9	2	3	5	1	8	7	4
7	4	5	2	8	6	3	1	9

Explanation: Each row, column and 3 x 3 box of the output matrix contains unique numbers.