Recursion and Backtracking

Important

What is Recursion?

- The process in which a function calls itself directly or indirectly is called recursion and the corresponding function is called as recursive function. Using recursive algorithm, certain problems can be solved quite easily.

Find Factorial

Find Fibionacci Series using recursion

Subset Sum : Sum of all Subsets

Problem Statement: Given an array print all the sum of the subset generated from it, in the increasing order.

Examples:

Example 1: Input: N = 3, arr[] = {5,2,1}
Output: 0,1,2,3,5,6,7,8
Explanation: We have to find all the subset's sum and print them.in this case the generated subsets are [[], [1], [2], [2,1], [5], [5,1], [5,2]. [5,2,1],so the sums we get will be 0,1,2,3,5,6,7,8

Input: N=3,arr[]= {3,1,2}
Output: 0,1,2,3,3,4,5,6
Explanation: We have to find all the subset's sum and print them.in this case the generated subsets are [[], [1], [2], [2,1], [3], [3,1], [3,2]. [3,2,1],so the sums we get will be 0,1,2,3,3,4,5,6

Subset – II | Print all the Unique Subsets

Problem Statement: Given an array of integers that **may contain duplicates** the task is to return all possible subsets. Return only **unique subsets** and they can be in any order.

Examples:

Example 1: Input: array[] = [1,2,2]
Output: [[ ],[1],[1,2],[1,2,2],[2],[2,2]]
Explanation: We can have subsets ranging from length 0 to 3. which are listed above. Also the subset [1,2] appears twice but is printed only once as we require only unique subsets.

Input: array[] = [1]
Output: [[ ], [1]]
Explanation: Only two unique subsets are available

Combination Sum – 1

Problem Statement:
Given an array of distinct integers and a **target**, you have to return *the list of all unique combinations where the chosen numbers sum to* target. You may return the combinations in any order.
The same number may be chosen from the given array an unlimited number of times. Two combinations are unique if the frequency of at least one of the chosen numbers is different.
It is guaranteed that the number of unique combinations that sum up to **target** is less than **150** combinations for the given input.

Examples:
Example 1: Input: array = [2,3,6,7], target = 7
Output: [[2,2,3],[7]]
Explanation: 2 and 3 are candidates, and 2 + 2 + 3 = 7. Note that 2 can be used multiple times. 7 is a candidate, and 7 = 7. These are the only two combinations.

Example 2: Input: array = [2], target = 1
Output: []
Explanation: No combination is possible.

Combination Sum II – Find all unique combinations

Problem Statement: Given a collection of candidate numbers (candidates) and a target number (target), find all unique combinations in candidates where the candidate numbers sum to target. Each number in candidates may only be used once in the combination.
Note: The solution set must not contain duplicate combinations.

Examples:
Example 1: Input: `candidates = [10,1,2,7,6,1,5], target = 8`
Output: `[ [1,1,6], [1,2,5], [1,7], [2,6]]`
Explanation: `These are the unique combinations whose sum is equal to target.`

Example 2: Input: `candidates = [2,5,2,1,2], target = 5`
Output: `[[1,2,2],[5]]`
Explanation: `These are the unique combinations whose sum is equal to target.`

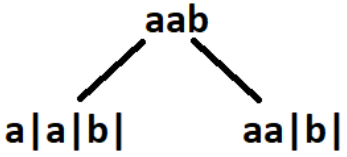
Palindrome Partitioning

Problem Statement: You are given a string s, partition it in such a way that every substring is a palindrome. Return all such palindromic partitions of s.
Note: A palindrome string is a string that reads the same backward as forward.

Examples:
Example 1: Input: s = “aab”
Output: [["a","a","b"], ["aa","b"]]

Explanation: The first answer is generated by making three partitions. The second answer is generated by making two partitions.

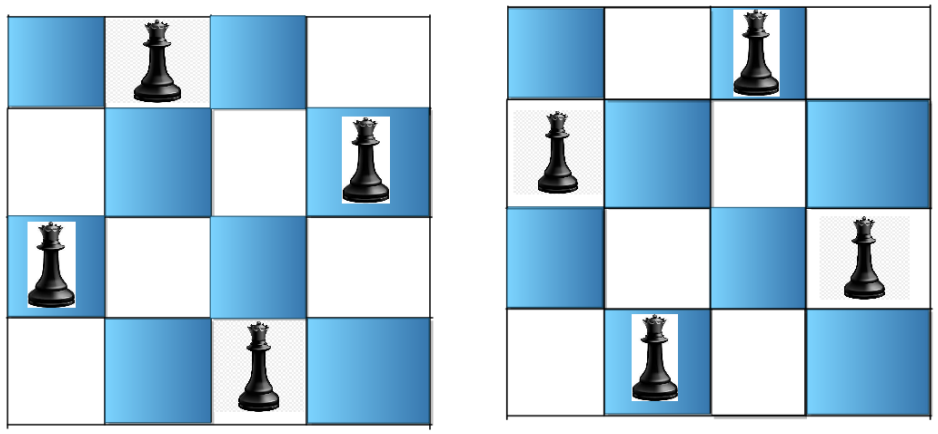
Example 2: Input: s = “aabb”
Output: [[“a”,”a”,”b”,”b”], [“aa”,”bb”], [“a”,”a”,”bb”], [“aa”,”b”,”b”]]
Explanation: See Figure



N Queen Problem | Return all Distinct Solutions to the N-Queens Puzzle

Problem Statement: The n-queens is the problem of placing n queens on n × n chessboard such that no two queens can attack each other. Given an integer n, return all distinct solutions to the n -queens puzzle. Each solution contains a distinct boards configuration of the queen's placement, where 'Q' and '.' indicate queen and empty space respectively.

Examples: Input: `n = 4`
Output: `[[".Q..","...Q","Q...","..Q."],["..Q.","Q...","...Q",".Q.."]]`
Explanation: There exist two distinct solutions to the 4-queens puzzle as shown below



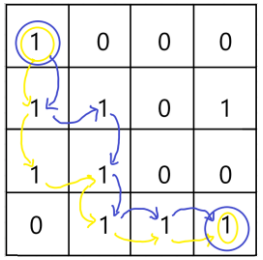
Two arrangements possible for 4 queens

Rat in a Maze

Rat in a Maze
Consider a rat placed at **(0, 0)** in a square matrix of order **N * N**. It has to reach the destination at **(N – 1, N – 1)**. Find all possible paths that the rat can take to reach from source to destination. The directions in which the rat can move are **'U'(up)**, **'D'(down)**, **'L' (left)**, **'R' (right)**. Value 0 at a cell in the matrix represents that it is blocked and the rat cannot move to it while value 1 at a cell in the matrix represents that rat can travel through it.
Note: In a path, no cell can be visited more than one time.
Print the answer in lexicographical(sorted) order

Examples:
Example 1: Input: N = 4 m[][] = {{1, 0, 0, 0}, {1, 1, 0, 1}, {1, 1, 0, 0}, {0, 1, 1, 1}}
Output: DDRDRR DRDDRR
Explanation: The rat can reach the destination at (3, 3) from (0, 0) by two paths - DRDDRR and DDRDRR, when printed in sorted order we get DDRDRR DRDDRR.

Example 2: Input: N = 2 m[][] = {{1, 0}, {1, 0}}
Output: No path exists and the destination cell is blocked.



Print All Permutations of a String/Array

Problem Statement: Given an array arr of distinct integers, print all permutations of String/Array.

Examples:

Example 1: 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: Given an array, return all the possible permutations.

Example 2: Input: arr = [0, 1]
Output: [[0, 1], [1, 0]]
Explanation: Given an array, return all the possible permutations.