BACKTRACKING

No.	Problem Statement	Solution	Time complexity	Space complexity
1	**Subsets**			
	Given an integer array nums of unique elements, return all possible subsets (the power set) Exp: nums = [1,2,3] Output: [[],[1],[2],[1,2],[3],[1,3],[2,3],[1,2,3]]	Idea: Brute Force (Recurion) - For each index 'i' 1) Consider the element at index 'i' 2) Dont consider the element at index 'i'	O(2^N)	O(N)
1	**Combination Sum**			
	Given distinct int array & a target, return list of all unique combos that sum to target. the same number can be chosen unlimited amount of time.	Idea: Brute Force - Generate all combinations 1) Consider ith element 2) Don't consider ith element	O(2^target/min) min = smallest element in the array	O(target/min)
2	**Word Search**			
	Given an m x n grid of characters board and a string word, return true if word exists in the grid. The word can be constructed from letters of horizontally and vertically adjacent cells. Same letter can't be used more than once.	Idea: Brute Force - Check in all the 4 directions - **Set the visited cells to '#' to avoid using the same letter more than once**	O(M*N*(4^L)) L = word.size()	O(L)
3				
4				
5				