**39. Combination Sum**

Given an array of **distinct** integers candidates and a target integer target, return *a list of all****unique combinations****of*candidates*where the chosen numbers sum to*target*.* You may return the combinations in **any order**.

The **same** number may be chosen from candidates 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.

**Input:** candidates = [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.

**40. Combination Sum II**

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.

**Input:** candidates = [10,1,2,7,6,1,5], target = 8

**Output:**

[

[1,1,6],

[1,2,5],

[1,7],

[2,6]

]

**78. Subsets**

Given an integer array nums of **unique** elements, return *all possible subsets (the power set)*.

The solution set **must not** contain duplicate subsets. Return the solution in **any order**.

**Input:** nums = [1,2,3]

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

**90. Subsets II**

Given an integer array nums that may contain duplicates, return *all possible subsets (the power set)*.

The solution set **must not** contain duplicate subsets. Return the solution in **any order**.

**Input:** nums = [1,2,2]

**Output:** [[],[1],[1,2],[1,2,2],[2],[2,2]]

**46. Permutations**

Given an array nums of distinct integers, return *all the possible permutations*. You can return the answer in **any order**.

**Input:** nums = [1,2,3]

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

**17. Letter Combinations of a Phone Number**

Given a string containing digits from 2-9 inclusive, return all possible letter combinations that the number could represent. Return the answer in **any order**.

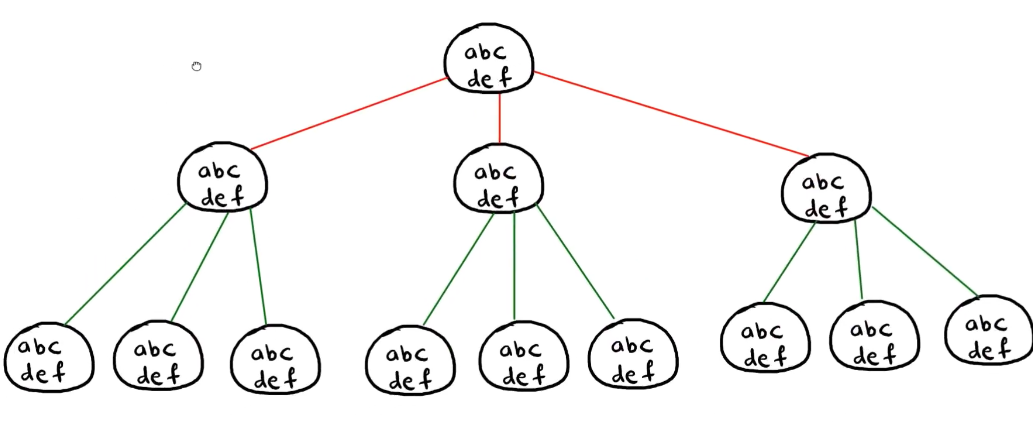
A mapping of digits to letters (just like on the telephone buttons) is given below. Note that 1 does not map to any letters.



**Input:** digits = "23"

**Output:** ["ad","ae","af","bd","be","bf","cd","ce","cf"]

**Sol:**



First we will take a map of character and string.

We will store its key number as key in map and string present on that key as value in the map.

Here digit is given as “23”

So we will take first 2 from it and find out map where key is 2. We get the key value pair ‘2’, “abc”

So current String is “abc”. We will go through for loop till its length.

Then we will add its ith index char to String builder so stringbuilder will contain “a” and increase the i to i+1

So now I becomes 0+1 = 1. And 1st index in digit is 3 and its corresponding value in map is “def”

Now again it will go through loop and its ith char will be added into string builder. So now String builder will contain “ad”.

Its length is 2 which is equal to digits length.

It means we got one answer. We will add this in result and return.

Once we return we delete the last char from the string builder and we will back track.

So stringbuilder contains now “a” . it will e from “def “ this time. Again its length = 2 and we will add this in result.

Similarly it will execute for other chars.