17. 电话号码的字母组合

题目：17. Letter Combinations of a Phone Number 语言: python3 英文版链接：https://leetcode.com/problems/letter-combinations-of-a-phone-number/description/ 中文版链接：https://leetcode-cn.com/problems/letter-combinations-of-a-phone-number/description/

题目分析

Backtracking（回溯）属于 DFS。

- 普通 DFS 主要用在 **可达性问题** ，这种问题只需要执行到特点的位置然后返回即可。
- 而 Backtracking 主要用于求解 **排列组合** 问题，例如有 { 'a','b','c' } 三个字符，求解所有由这三个字符排列得到的字符串，这种问题在执行到特定的位置返回之后还会继续执行求解过程。

因为 Backtracking 不是立即就返回，而要继续求解，因此在程序实现时，需要注意对元素的标记问题：

- 在访问一个新元素进入新的递归调用时，需要将新元素标记为已经访问，这样才能在继续递归调用时不用重复访问该元素；
- 但是在递归返回时，需要将元素标记为未访问，因为只需要保证在一个递归链中不同时访问一个元素，可以访问已经访问过但是不在当前递归链中的元素。

答案

```
class Solution:
    def letterCombinations(self, digits: str) -> List[str]:
        self.keys=["", "", "abc", "def", "ghi", "jkl", "mno", "pqrs", "tuv", "wxyz"]
        combinations, prefix = [], ""
        if not digits:
            return combinations
        self.doCombinations(prefix, combinations, digits)
        return combinations

    def doCombinations(self, prefix: str, combinations: List[str], digits: str):
        if len(prefix) == len(digits):
            combinations.append(prefix)
            return
        for i in self.keys[int(digits[len(prefix)])]:
            prefix += i
            self.doCombinations(prefix, combinations, digits)
            prefix = prefix[:-1]
```