2에서 9까지 숫자가 주어졌을 때 전화 번호로 조합 가능한 모든 문자를 출력하라.

Example 1:
Input: digits = "23"
Output: ["ad","ae","af","bd","be","bf","cd","ce","cf"]

1.브루트포스

```
class Solution:
    def letterCombinations(self, digits: str) -> List[str]:
        def comb(dest: str, string: str) -> None:
            if len(dest) == 0:
                if len(string) > 0:
                    result.append(string)
                return
            cur = dest[0]
            dest = dest[1:]
            for s in strs[cur]:
                comb(dest[:], (string + s)[:])

        strs = {}
        strs['2'] = 'abc'
        strs['3'] = 'def'
        strs['4'] = 'ghi'
        strs['5'] = 'jkl'
        strs['6'] = 'mno'
        strs['7'] = 'pqrs'
        strs['8'] = 'tuv'
        strs['9'] = 'wxyz'
        result = []
        comb(digits, '')
        return result
```

```
class Solution:
    def letterCombinations(self, digits: str) -> List[str]:
        def dfs(index, path):
            if len(path) == len(digits):
                result.append(path)
                return
            for i in range(index, len(digits)):
                for j in dic[digits[i]]:
                    dfs(i + 1, path + j)

        if not digits:
            return []
        dic = {
            '2': 'abc',
            '3': 'def',
            '4': 'ghi',
            '5': 'jkl',
            '6': 'mno',
            '7': 'pqrs',
            '8': 'tuv',
    '9': 'wxyz'
}
result = []
dfs(0, '')
return result
```