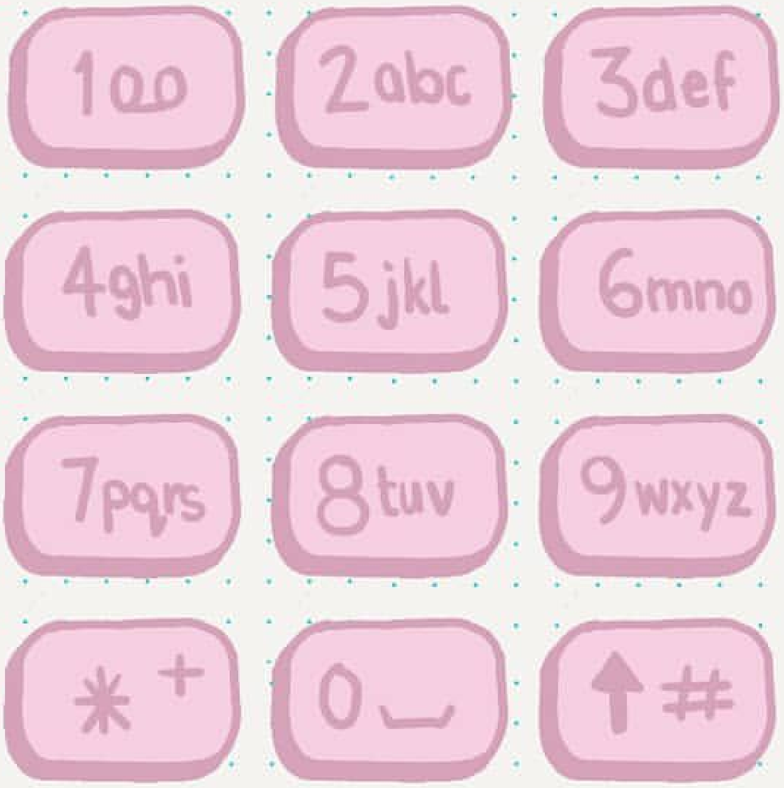
leetcode



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

THE CODE DOSE

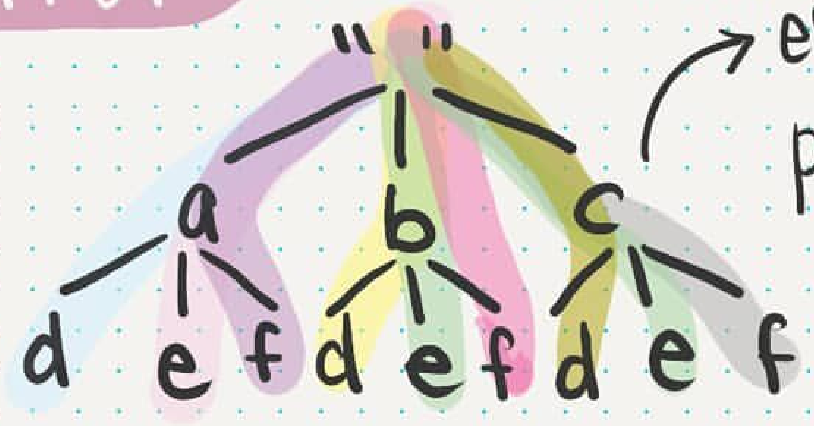
EXAMPLE

INPUT

"23"

given number

OUTPUT



every possible path from root to leaf

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

THE CODE DOSE

SOLUTION

```python
d = {
    "2": ["a", "b", "c"],
    "3": ["d", "e", "f"],
    "4": ["g", "h", "i"],
    "5": ["j", "k", "l"],
    "6": ["m", "n", "o"],
    "7": ["p", "q", "r", "s"],
    "8": ["t", "u", "v"],
    "9": ["w", "x", "y", "z"],
}

def helper(digits, i, comb, res):
    if i >= len(digits):
        if not len(digits) == 0:
            res.append(comb)
        return
    for c in d[digits[i]]:
        helper(digits, i+1, comb + c, res)

class Solution:
    def letterCombinations(self, digits: str) -> List[str]:
        res = []
        helper(digits, 0, "", res)
        return res
```

create the number to character mappings

this means we reached the end of a path

add this path to res list

branch

start with empty string i.e. root of tree

THE CODE DOSE