

Letter Combinations of a Phone Number

# Index	17
📅 CreatedAt	@September 28, 2022
👤 Person	A Ally Hyeseong Kim
☀️ Status	Done
🏷️ Tags	Graph Python
📅 UpdatedAt	@September 28, 2022

References

LeetCode - The World's Leading Online Programming Learning Platform

Level up your coding skills and quickly land a job. This is the best place to expand your knowledge and get prepared for your next interview.

🔗 <https://leetcode.com/problems/letter-combinations-of-a-phone-number/>



파이썬 알고리즘 인터뷰

2021 세종도서 학술부문 선정작. 현업과 실무에 유용한 주요 알고리즘 이론을 깊숙이 이해하고, 파이썬의 핵심 기능과 문법까지 상세하게 이해할 수 있는 취업용 코딩 테스트를 위한 완벽 가이드다. 200여 개가 넘는...

🔗 <https://www.aladin.co.kr/shop/wproduct.aspx?ItemId=245495826>



References

1. BFS
2. DFS

1. BFS

```
class Solution:
    def letterCombinations(self, digits: str) -> List[str]:
        digit_dict = {
            "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"]
        }

        answer = []
        for d in digits:
            if not answer:
                answer = digit_dict[d]
            else:
                prev = answer
                answer = []
                for d in digit_dict[d]:
```

```
        answer += [p + d for p in prev]

    return answer
```

2. DFS

```
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)

        dic = {"2": "abc", "3": "def", "4": "ghi", "5": "jkl", "6": "mno", "7": "pqrs", "8": "tuv", "9": "wxyz"}
        result = []
        dfs(0, "")

        return result
```