

Permutations II

Leetcode 47

202021018 오윤주

문제

47. Permutations II

Medium



4243



88



Add to List



Share

Given a collection of numbers, `nums`, that might contain duplicates, return *all possible unique permutations* **in any order**.

Example 1:

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

Output:

```
[[1,1,2],  
 [1,2,1],  
 [2,1,1]]
```

Example 2:

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

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

구현

```
class Solution:
    def permuteUnique(self, nums: List[int]) -> List[List[int]]:
        li = []
        nums.sort()
        // 정렬

        def dfs(i, nums):
            if i >= len(nums):
                li.append(nums) // 원래 입력받은 수
                return

            for j in range(i, len(nums)):
                if j > i and nums[i] == nums[j]: // 중복 삭제
                    continue

                nums[i], nums[j] = nums[j], nums[i] // 자리 바꿔주기
                dfs(i+1, nums.copy())

        dfs(0, nums)

        return li
```

감사합니다.