Problem 47: Permutations II

Problem Information

Difficulty: Medium
Acceptance Rate: 62.51%
Paid Only: No
Tags: Array, Backtracking, Sorting

Problem Description

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]]

Constraints:

* `1 <= nums.length <= 8` * `-10 <= nums[i] <= 10`

Code Snippets

C++:

```
class Solution {
public:
vector<vector<int>> permuteUnique(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public List<List<Integer>> permuteUnique(int[] nums) {

}
}
```

Python3:

```
class Solution:
def permuteUnique(self, nums: List[int]) -> List[List[int]]:
```