

Problem 90: Subsets II

Problem Information

Difficulty: Medium

Acceptance Rate: 60.44%

Paid Only: No

Tags: Array, Backtracking, Bit Manipulation

Problem Description

Given an integer array `nums` that may contain duplicates, return all possible subsets (the power set).

The solution set **must not** contain duplicate subsets. Return the solution in **any order**.

Example 1:

Input: nums = [1,2,2] **Output:** [[],[1],[1,2],[1,2,2],[2],[2,2]]

Example 2:

Input: nums = [0] **Output:** [[],[0]]

Constraints:

$1 \leq \text{nums.length} \leq 10$ $-10 \leq \text{nums}[i] \leq 10$

Code Snippets

C++:

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

```
}  
};
```

Java:

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

Python3:

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