Problem 78: Subsets

Problem Information

Difficulty: Medium
Acceptance Rate: 81.68%
Paid Only: No
Tags: Array, Backtracking, Bit Manipulation

Problem Description

Given an integer array `nums` of **unique** elements, 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 = [1,2,3] **Output:** [[],[1],[2],[1,2],[3],[1,3],[2,3],[1,2,3]]

Example 2:

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

Constraints:

* `1 <= nums.length <= 10` * `-10 <= nums[i] <= 10` * All the numbers of `nums` are **unique**.

Code Snippets

C++:

```
class Solution {
public:
vector<vector<int>> subsets(vector<int>& nums) {
}
};
```

Java:

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

}
}
```

Python3:

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