

78. Subsets

Solved

Medium Topics Companies

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**.

The screenshot shows a code editor with the following details:

- Left Pane (Performance):**
 - Status: Accepted
 - Submitted by: ZhukovIya at Jul 06, 2024 20:37
 - Runtime: 107 ms | Beats 25.96%
 - Memory: 45.97 MB | Beats 33.96%
 - A bar chart showing runtime distribution across various time intervals.
- Right Pane (Code):**

```
1 using System;
2 using System.Collections.Generic;
3
4 public class Solution {
5     public IList<IList<int>> Subsets(int[] nums) {
6         var results = new List<IList<int>>();
7         GenerateSubsets(nums, 0, new List<int>(), results);
8         return results;
9     }
10
11     private void GenerateSubsets(int[] nums, int index, List<int> current, List<IList<int>> results)
```

Код:

```
using System;
using System.Collections.Generic;

public class Solution
{
    public IList<IList<int>> Subsets(int[] nums)
    {
```

```
        var results = new List<IList<int>>();
        GenerateSubsets(nums, 0, new List<int>(), results);
        return results;
    }

    private void GenerateSubsets(int[] nums, int index, List<int> current,
List<IList<int>> results)
    {
        results.Add(new List<int>(current));

        for (int i = index; i < nums.Length; i++)
        {
            current.Add(nums[i]);
            GenerateSubsets(nums, i + 1, current, results);
            current.RemoveAt(current.Count - 1);
        }
    }
}
```