

Description

Editorial

Solutions (8.5K)

Submissions

## 78. Subsets

Medium

15.4K

223



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:

- $1 \leq \text{nums.length} \leq 10$
- $-10 \leq \text{nums}[i] \leq 10$
- All the numbers of `nums` are **unique**.

i Java

Auto

```
1 class Solution {
2     public List<List<Integer>> subsets(int[] nums) {
3         List<List<Integer>> res = new ArrayList<>();
4         res.add(new ArrayList<>());
5
6         for(int i : nums){
7             int n = res.size();
8             for(int j = 0; j < n; j++){
9                 List<Integer> ans = new ArrayList<>(res.get(j));
10                ans.add(i);
11                res.add(ans);
12            }
13        }
14        return res;
15    }
16 }
```

Testcase

Result

**Accepted** Runtime: 0 ms

• Case 1 • Case 2

Input

nums =

[1,2,3]

Output

[[], [1], [2], [1,2], [3], [1,3], [2,3], [1,2,3]]

Expected

Console



Run

Submit

## 518. Coin Change II

Medium 8.2K 139

Companies

You are given an integer array `coins` representing coins of different denominations and an integer `amount` representing a total amount of money.

Return the number of combinations that make up that amount. If that amount of money cannot be made up by any combination of the coins, return 0.

You may assume that you have an infinite number of each kind of coin.

The answer is **guaranteed** to fit into a signed 32-bit integer.

### Example 1:

**Input:** amount = 5, coins = [1,2,5]

**Output:** 4

**Explanation:** there are four ways to make up the amount:

5=5

5=2+2+1

5=2+1+1+1

5=1+1+1+1+1

### Example 2:

**Input:** amount = 3, coins = [2]

**Output:** 0

**Explanation:** the amount of 3 cannot be made up just with coins of 2.

i Java Auto

```
1 public class Solution {
2     public int change(int amount, int[] coins) {
3         int[] dp = new int[amount + 1];
4         dp[0] = 1;
5
6         for (int coin : coins) {
7             for (int j = coin; j <= amount; j++) {
8                 dp[j] += dp[j - coin];
9             }
10        }
11
12        return dp[amount];
13    }
14 }
```

Testcase Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

amount =

5

coins =

[1,2,5]

Output

Console



Run

Submit