Subsets

Try to solve the Subsets problem.

We'll cover the following

Statement

Given an array of integers, `nums`, find all possible subsets of `nums`, including the empty set.

> **Note:** The solution set must not contain duplicate subsets. You can return the solution in any order.

Constraints:

- $1 \leq$ `nums.length` ≤ 10
- $-10 \leq$ `nums[i]` ≤ 10
- All the numbers of `nums` are unique.

Examples

Sample example 1

Input

3	6	9

Output

[]	[3]	[6]	[9]	[3, 6]	[3, 9]	[6, 9]	[3, 6, 9]

All possible subsets are given above.

1 of 4

Understand the problem

Let's take a moment to make sure you've correctly understood the problem. The quiz below helps you check if you're solving the correct problem:

Subsets

1 Find all subsets for the given set.

nums = [6, 7, 8]

A) [[], [6], [7], [8], [6, 7], [6, 8], [7, 8]]

B) [[], [6], [7], [8], [6, 7], [6, 8], [7, 8], [6, 7, 8]]

C) [[], [6], [6, 7], [6, 8], [7, 8], [6, 7, 8]]

D) [[6], [7], [8], [6, 7], [6, 8], [7, 8], [6, 7, 8]]

Submit Answer

Question 1 of 4
0 attempted

Reset Quiz

Figure it out!

We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.

Drag and drop the cards to rearrange them in the correct sequence.

Compute the number of possible subsets of the given set using 2^n, where n is the number of elements.

Start a loop from 0 to the count of subsets and add an empty list to the results list in the first iteration.

In each iteration, create a bit mask of length n for each element in the input set. If

the i^{th} bit is set, $set[i]$ will be present in the current subset.

After iterating over all elements in the input set, append the current subset to the list of subsets.

Try it yourself

Implement your solution in the following coding playground:

Java

usercode > FindSubsets.java

```java
import java.util.*;

class FindSubsets {

    public static List<List<Integer>> findAllSubsets(int[] nums) {
        // Write your code here

        // Your code will replace the below return placeholder

        List<List<Integer>> setsList = new ArrayList<>();
        return setsList;
    }
}
```

Powered by AI

Submit

Test Cases | Results

Case 1 | Case 2 | Case 3

Input #1

[1]

Subsets

Back

Subsets: Introduction

Next

Solution: Subsets

Mark as Completed