Our Solution(s)   Run Code

Solution 1   Solution 2

```swift
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
    // O(n * n!) time | O(n * n!) space
    func permutations(array: inout [Int]) -> [[Int]] {
        var permutations = [[Int]]()
        permutationsHelper(firstIndex: 0, array: &array, permut

        return permutations
    }

    func permutationsHelper(firstIndex: Int, array: inout [Int]
        if firstIndex == array.count - 1 {
            permutations.append(array)
        } else {
            for secondIndex in firstIndex ..< array.count {
                swap(firstIndex: firstIndex, secondIndex: secon
                permutationsHelper(firstIndex: firstIndex + 1, 
                swap(firstIndex: firstIndex, secondIndex: secon
            }
        }
    }

    func swap(firstIndex: Int, secondIndex: Int, array: inout [I
        let temp = array[firstIndex]
        array[firstIndex] = array[secondIndex]
        array[secondIndex] = temp
    }
}
```

Your Solutions   Run Code

Solution 1   Solution 2   Solution 3

```swift
class Program {
    func permutations(array: inout [Int]) -> [[Int]] {
        // Write your code here.
        return []
    }
}
```

Our Tests   Custom Output   Submit Code

```
func test() {
    let program = Program()
    context("Test case 1") { } in Stmps in Void in
        var firstTest: [Int] = []
        var firstTestResults = program.permutations(string: )
        try assertEqual(0, firstTestResults.count)
    }

    context("Test case 2") { } in Stmps in Void in
        var secondTest: [Int] = [a]
        var secondTestResults = program.permutations(string: )
        try assertEqual(0, secondTestResults.count)
        try assertT(secondTestResults.contains(a))
    }

    context("Test case 3") { } in Stmps in Void in
        var thirdTest: [Int] = [a, b]
        var thirdTestResults = program.permutations(string: )
        try assertEqual(0, thirdTestResults.count)
        try assertT(thirdTestResults.contains(a, b))
        try assertT(thirdTestResults.contains(b, a))
    }
    context("Test case 4") { } in Stmps in Void in
```

Run or submit code when you're ready.