

# Permutations II

[My Submissions \(/problems/permutations-ii/submissions/\)](/problems/permutations-ii/submissions/)

Total

Question

Solution

Accepted:

**45990** Total Submissions: **177562** Difficulty: **Medium**

Given a collection of numbers that might contain duplicates, return all possible unique permutations.

For example,

[1,1,2] have the following unique permutations:

[1,1,2] , [1,2,1] , and [2,1,1] .

[Show Tags](#)[Show Similar Problems](#)

Have you met this question in a real interview?

[Discuss \(/discuss/questions/oj/permutations-ii\)](/discuss/questions/oj/permutations-ii)

Java ▼



```
1 public class Solution {
2     public List<List<Integer>> permuteUnique(int[] nums) {
3         Arrays.sort(nums);
4         List<List<Integer>> returnColumn = new ArrayList<List<Integer>>();
5         Integer[] prefix = new Integer[nums.length];
6         int[] visited = new int[nums.length];
7         DFS(nums,0,prefix,visited,returnColumn);
8         return returnColumn;
9     }
10    private void DFS(int[] nums,int step,Integer[] prefix,int[] visited,List<
11        if(step >= nums.length){
12            returnColumn.add(Arrays.asList(Arrays.copyOf(prefix,prefix.length)
13            return;
14        }
15        for(int i=0;i<nums.length;i++){
16            if(visited[i] == 1)continue;
17            if(i>0 && nums[i] == nums[i-1] && visited[i-1] ==0)continue;
18            visited[i] = 1;
19            prefix[step] = nums[i];
20            DFS(nums,step+1,prefix,visited,returnColumn);
21            visited[i] = 0;
22        }
23    }
24 }
```

✉ Send Feedback (mailto:admin@leetcode.com?subject=Feedback)

Custom Testcase ☐

Run Code

Submit Solution

[Frequently Asked Questions \(/faq/\)](#) | [Terms of Service \(/tos/\)](#)

[Privacy](#)

Copyright © 2015 LeetCode