47. Permutations II

Add to List | Question | Editorial Solution | My Submissions (/problems/permutations-ii/submissions/)

Total Accepted: **102788** Total Submissions: **330119** Difficulty: **Medium** Contributors: **Admin**

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],
  [2,1,1]
]
```

Notes

Hide Company Tags | LinkedIn (/company/linkedin/) | Microsoft (/company/microsoft/)

Hide Tags | Backtracking (/tag/backtracking/)

Hide Similar Problems | (M) Next Permutation (/problems/next-permutation/) | (M) Permutations (/problems/permutations/) | (M) Palindrome Permutation II (/problems/palindrome-permutation-ii/)

Have you met this question in a real interview? Yes No

Discuss (https://discuss.leetcode.com/category/55) | Top Solutions | Pick One (/problems/random-one-question/)

C++

```cpp
class Solution {
public:
    vector<vector<int>> permuteUnique(vector<int>& nums) {
        vector<vector<int>> res;
        sort(nums.begin(),nums.end());
        res.push_back(nums);
        int n = nums.size();
        if(n==1) return res;
        int j,k,l;
        while(true){
            j = n-2;
            while(j>=0 && nums[j]>=nums[j+1]) j--;
            if(j<0) break;
            l = n-1;
            while(nums[j]>=nums[l]) l--;
            swap(nums[j],nums[l]);
            for(k=j+1,l=n-1;k<l;k++,l--)
                swap(nums[k],nums[l]);
            res.push_back(nums);
        }
        return res;
    }
};
```

Custom Testcase

Contribute Testcase

Run Code | Submit Solution