

[LintCode领扣题解 \(/problem\)](#) / 带重复元素的排列 · Permutations II

## 带重复元素的排列 · Permutations II

中文

[微软 \(/problem/?tags=microsoft\)](#)[领英 \(/problem/?tags=linkedin\)](#)[Depth-first Search \(/problem/?tags=depth-first-search\)](#)[递归 \(/problem/?tags=recursion\)](#)

### 描述

给出一个具有重复数字的列表, 找出列表所有不同的排列。

### 样例

样例 1:

```
输入: [1,1]
输出:
[
  [1,1]
]
```

样例 2:

```
输入: [1,2,2]
输出:
[
  [1,2,2],
  [2,1,2],
  [2,2,1]
]
```

### 挑战

使用递归和非递归分别完成该题。

在线评测地址: <https://www.lintcode.com/problem/permutations-ii/> (<https://www.lintcode.com/problem/permutations-ii/>)

收起题目描述 ^

语言类型

[ALL \(23\)](#)[python \(11\)](#)[java \(6\)](#)[cpp \(6\)](#)[上传题解](#)

令狐冲

更新于 12/13/2020, 1:14:58 AM

使用排列式深度优先搜索算法。

和没有重复元素的 Permutation 一题相比, 只加了两句话:

1. `Arrays.sort(nums)` // 排序这样所有重复的数
2. `if (i > 0 && numsi () == numsi - 1 () && !visitedi - 1 ()) { continue; }` // 跳过会造成重复的情况

```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
public class Solution {
    /**
     * @param : A list of integers
     * @return: A list of unique permutations
     */
    public List<List<Integer>> permuteUnique(int[] nums) {
        List<List<Integer>> results = new ArrayList<>();
        if (nums == null) {
            return results;
        }

        Arrays.sort(nums);
        dfs(nums, new boolean[nums.length], new ArrayList<Integer>(), results);

        return results;
    }

    private void dfs(int[] nums,
                     boolean[] visited,
                     List<Integer> permutation,
                     List<List<Integer>> results) {
        if (nums.length == permutation.size()) {
            results.add(new ArrayList<Integer>(permutation));
            return;
        }

        for (int i = 0; i < nums.length; i++) {
            if (visited[i]) {
                continue;
            }
            if (i > 0 && nums[i] == nums[i - 1] && !visited[i - 1]) {
                continue;
            }

            permutation.add(nums[i]);
            visited[i] = true;
            dfs(nums, visited, permutation, results);
            visited[i] = false;
            permutation.remove(permutation.size() - 1);
        }
    }
};

```

👍 获赞 8

💬 8 条评论

# 你的口袋题库

## 2000+ 算法真题、国内外名企题库免费开放



九章算法APP

九章-小原

更新于 12/23/2020, 10:01:49 PM

## 解题思路

- 这道题我们需要使用dfs+回溯的方法来进行求解。
- 我们定义 dfs 函数，使用递归的方法对决策树进行深度优先遍历。对于长度为 n 的数组 nums，我们一位一位地生成它的排列数组，每深入一层数组长度就加1，遍历到叶节点时生成数组的长度达到 n，即为我们的答案。
- 由于数组中有重复元素，所以我们在遍历时需要剪枝操作。

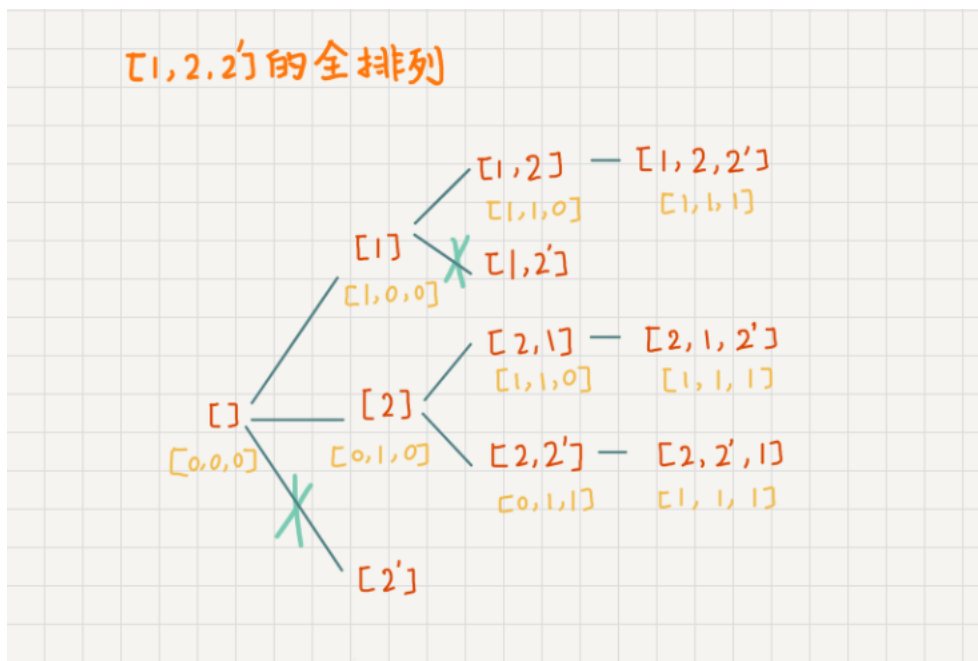
## 算法流程

- 首先对数组进行排序，以使得重复元素相邻，这样才能进行剪枝。
- 定义数组 used，used[i] 表示 nums[i] 是否已使用过，初始化全为 false。数组 path，表示从根结点到该节点经过的路径，即当前已生成的数组，初始化为空。数组 res 存储结果。
- 使用 dfs 函数进行递归遍历
  - 递归出口：如果 path 的长度与 nums 的长度相等，说明已经生成好了排列数组 path，那么我们把它的拷贝加入 res 中。
  - 遍历 nums 中的每个元素，对于 nums[i]
    - 如果 path 中已经存在，即 used[i] 为 true，跳过
    - 如果它和前一位元素相等，即  $\text{nums}[i-1] == \text{nums}[i]$ ，并且前一位元素已经搜索并回溯过了，即  $!used[i-1]$ ，为了避免生成重复的排列数组，也跳过
    - 排除上述两种情况后，把 nums[i] 变为 true，然后对新生成的 path 继续送入 dfs 函数中。
    - 最后进行回溯操作，即删除 path[i]，used[i] 变为 false。



## 举例说明

- 如图所示，nums = [1, 2, 2]，第二个2标记为2'用于区分相同元素。每个节点有 path 和 used 两个属性。
- 首先，在根结点，path 为 []，used 全为 false（图中标为 [0, 0, 0]）。然后进行dfs遍历，到下一层，先加入元素1，path 为 [1]，used 为 [1, 0, 0]。再到下一层，由于1已经使用过了，我们加入元素2，path 为 [1, 2]，used 为 [1, 1, 0]。这样，每深一层 path 长度加1。达到最底层的叶节点，path 为 [1, 2, 2]，把它加入 res 中。同理，可以得到其他的叶节点。
- 注意，图中标出画叉的地方，代表出现了重复元素而进行剪枝。



## 复杂度分析

- 时间复杂度： $O(n \times n!)$ ，这里  $n$  为数组的长度。当没有重复元素时，排列数组有  $n!$  个，即最深层有  $n!$  个叶子节点，而拷贝操作需要  $n$ ，所以时间复杂度为  $O(n \times n!)$
- 空间复杂度： $O(n \times n!)$ 。最差情况下，返回的全排列数组有  $n!$  个，每个长度为  $n$ 。

# 代码

python

C++

java

```
/**
 * 本参考程序由九章算法用户提供。版权所有，转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作，授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括：九章算法班 2020升级版，算法强化班，算法基础班，北美算法面试高频题班，Java 高级工程师 P6+ 小班课，面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括：系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括：动态规划专题班，Big Data - Spark 项目实战，Django 开发项目课
 * - 更多详情请见官方网站：http://www.jiuzhang.com/?utm_source=code
 */
class Solution:
    """
    @param : A list of integers
    @return: A list of unique permutations
    """
    def permuteUnique(self, nums):
        res = []
        used = [False] * len(nums)
        path = []
        # 排序
        nums = sorted(nums)
        # dfs
        self.dfs(nums, used, path, res)
        return res
    def dfs(self, nums, used, path, res):
        # 叶子节点
        if len(path) == len(nums):
            res.append(path[:])
            return
        # 非叶节点
        for i in range(len(nums)):
            # 元素已访问过 或者 是重复元素
            if used[i] or (i > 0 and nums[i] == nums[i-1] and not used[i-1]):
                continue
            # 在路径添加该节点，递归
            used[i] = True
            self.dfs(nums, used, path + [nums[i]], res)
            # 回溯
            used[i] = False
```

👍 获赞 5

💬 4 条评论



令狐冲

更新于 12/31/2020, 5:46:10 AM

python

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ /
Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
class Solution:
    """
    @param nums: A list of integers.
    @return: A list of unique permutations.
    """
    def permuteUnique(self, nums):
        # write your code here
        def _permute(result, temp, nums):
            if nums == []:
                result += [temp]
            else:
                for i in range(len(nums)):
                    if i > 0 and nums[i] == nums[i-1]:
                        continue
                    _permute(result, temp + [nums[i]], nums[:i] + nums[i+1:])
        if nums is None:
            return []

        if len(nums) == 0:
            return [[]]

        result = []
        _permute(result, [], sorted(nums))
        return result
```

👍 获赞 0

💬 3 条评论



令狐冲

更新于 12/31/2020, 5:45:52 AM

cpp

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution {
private:
    void helper(vector<vector<int> > &results,
                vector<int> &permutation,
                vector<int> &nums,
                bool used[]) {
        if (nums.size() == permutation.size()) {
            results.push_back(permutation);
            return;
        }

        for (int i = 0; i < nums.size(); i++) {
            if (used[i]) {
                continue;
            }
            if (i > 0 && used[i - 1] == false && nums[i] == nums[i-1]) {
                continue;
            }

            used[i] = true;
            permutation.push_back(nums[i]);
            helper(results, permutation, nums, used);
            permutation.pop_back();
            used[i] = false;
        }
    }

public:
    vector<vector<int> > permuteUnique(vector<int> &nums) {
        vector<vector<int> > results;
        vector<int> permutation;
        bool used[nums.size()];

        for (int i = 0; i < nums.size(); i++) {
            used[i] = false;
        }

        sort(nums.begin(), nums.end());

        helper(results, permutation, nums, used);
        return results;
    }
};
```

👍 获赞 0    💬 添加评论



华山扫地僧

更新于 6/9/2020, 7:03:47 AM

使用九章算法班模板进行解决, 带回溯的DFS + Visited

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
class Solution:
    """
    @param : A list of integers
    @return: A list of unique permutations
    """

    def permuteUnique(self, nums):
        self.results = []
        self.visited = {i: False for i in range(len(nums))}
        self.dfs([], sorted(nums))
        return self.results

    def dfs(self, path, nums):
        if len(path) == len(nums):
            self.results.append(path[:])
            return

        for i in range(len(nums)):
            if self.visited[i]:
                continue

            if i != 0 and nums[i] == nums[i - 1] and self.visited[i - 1]:
                continue

            self.visited[i] = True
            path.append(nums[i])
            self.dfs(path, nums)
            path.pop()
            self.visited[i] = False
```

👍 获赞 4

💬 1 条评论



九章用户JQEMK1

更新于 6/9/2020, 7:03:47 AM

dfs + 剪枝。1.简单的一句话的剪枝方法: 判断该数是否在之前的数组里面出现过。2.对原数组排序后的剪枝

```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution:
    """
    @param : A list of integers
    @return: A list of unique permutations
    """

    def permuteUnique(self, nums):
        # write your code here
        if not nums:
            return [[]]
        results = []
        self.dfs(results, [], nums)
        return results

    def dfs(self, results, path, nums):
        if not nums:
            results.append(path[:])
            return
        for i in range(len(nums)):
            if nums[i] not in nums[:i]:
                path.append(nums[i])
                self.dfs(results, path, nums[:i] + nums[i+1:])
                path.pop()

class Solution:
    """
    @param : A list of integers
    @return: A list of unique permutations
    """

    def permuteUnique(self, nums):
        # write your code here
        result = []
        s = sorted(nums)
        self.dfs(result, [], s)
        return result

    def dfs(self, result, path, s):
        if not s:
            result.append(path[:])
            return
        for i in range(len(s)):
            if i != 0 and s[i] == s[i-1]: continue
            path.append(s[i])
            self.dfs(result, path, s[:i]+s[i+1:])
            path.pop()

```

👍 获赞 4

💬 添加评论



同学

更新于 6/9/2020, 7:03:50 AM

与dfs基于组合的类似, 只是加一个数组记录是否访问过就好了。



```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution:
    """
    @param : A list of integers
    @return: A list of unique permutations
    """

    def permuteUnique(self, nums):
        # write your code here
        if len(nums) == 0:
            return [[]]
        res = []
        nums.sort()
        visited = [[False] for i in range(len(nums))]
        self.dfs(nums, [], res, visited)
        return res

    def dfs(self, nums, subset, result, visited):
        if len(nums) == len(subset):
            result.append(subset[:])

        for i in range(len(nums)):
            if visited[i] == True:
                continue
            if i-1 >= 0 and visited[i-1] == True and nums[i-1] == nums[i]:
                continue
            visited[i] = True
            subset.append(nums[i])
            self.dfs(nums, subset, result, visited)
            visited[i] = False
            subset.pop()

```

👍 获赞 2

💬 添加评论

**DIng**

更新于 6/9/2020, 7:03:56 AM

利用nextPermutation的解法。不能对int数组使用 Arrays.asList()。

```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
public class Solution {
    /**
     * @param : A list of integers
     * @return: A list of unique permutations
     */
    public List<List<Integer>> permuteUnique(int[] nums) {
        // write your code here
    }
}

```

```
List<List<Integer>> results = new ArrayList<>();
if (nums == null) {
    return results;
}

Arrays.sort(nums);

getAllPermutation(results, nums);
return results;
}

private void getAllPermutation(List<List<Integer>> results,
                               int[] nums) {

    results.add(getList(nums));

    while (nextPermutation(nums)) {
        results.add(getList(nums));
    }
}

private boolean nextPermutation(int[] nums) {
    int i = 0;
    for (i = nums.length - 1; i > 0; i--) {
        if (nums[i] > nums[i - 1]) {
            break;
        }
    }

    if (i < 1) {
        return false;
    }

    int j = 0;
    for (j = nums.length - 1; j > i - 1; j--) {
        if (nums[j] > nums[i - 1]) {
            reverseTwo(i - 1, j, nums);
            break;
        }
    }

    reverseAll(i, nums);

    return true;
}

private void reverseTwo(int i, int j, int[] nums) {
    int temp = nums[i];
    nums[i] = nums[j];
    nums[j] = temp;
}

private void reverseAll(int start, int[] nums) {
    for (int i = start, j = nums.length - 1; i < j; i++, j--) {
        reverseTwo(i, j, nums);
    }
}

private List<Integer> getList(int[] nums) {
    List<Integer> list = new ArrayList<>();
    for (int val : nums) {
        list.add(val);
    }
    return list;
}
};
```

👍 获赞 1    💬 添加评论



九章用户FQI8G1

更新于 6/9/2020, 7:03:55 AM

Python version of non-recursion: Use cur stack to record the cur substring and mark to record the location visited in nums.

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
class Solution:
    def permuteUnique(self, nums):
        if not nums:
            return []
        nums.sort()
        res = []
        cur = [i for i in nums]
        visited = [i for i in range(len(nums))]
        print(visited, cur)
        while cur:
            print("here")
            head = cur.pop()
            mark = visited.pop()
            if len(head) == len(nums) and head not in res:
                res.append(list(head))
                continue
            for i in range(len(nums)):
                if i not in mark:
                    cur.append(list(head+[nums[i]]))
                    visited.append(list(mark+[i]))
        return res
```

👍 获赞 1    💬 添加评论



jerron

更新于 6/9/2020, 7:03:53 AM

dfs recursive & non recursive skip duplicates within same level

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
//solution 1: recursive
class Solution {
public:
    /**
     * @param : A list of integers
     * @return: A list of unique permutations
     */
```

```

vector<vector<int>> permuteUnique(vector<int> &nums) {
    // write your code here
    sort(nums.begin(),nums.end());
    vector<remove_reference<decltype(nums)>::type> result;
    function<decltype(result)(decltype(nums))> dfs;
    dfs=[&dfs](decltype(nums)nums)->decltype(result){
        if(nums.empty())
            return {{}};
        decltype(result) res;
        for(auto i=0;i<nums.size();++i)
            if(i<1 || nums[i]!=nums[i-1]){
                auto n=nums;
                n.erase(n.begin()+i);
                for(auto r:dfs(n)){
                    res.push_back(r);
                    res.back().push_back(nums[i]);
                }
            }
        return res;
    };
    return dfs(nums);
}

//solution 2: non recursive
class Solution {
public:
    /*
    * @param : A list of integers
    * @return: A list of unique permutations
    */
    vector<vector<int>> permuteUnique(vector<int> &nums) {
        // write your code here
        int s=nums.size();
        if(s<1)
            return {{}};
        vector<int> states(s,0);
        vector<bool> used(s+1,false);
        vector<vector<int>> result;
        sort(nums.begin(),nums.end());
        nums.insert(nums.begin(),INT_MIN);
        for(int i=0;i>=0;){
            auto prev=states[i];
            auto ok=false;
            for(used[prev]=false;!ok&&++states[i]<=s;
                ok=!used[states[i]]&&nums[states[i]]!=nums[prev]);
            if(ok){
                used[states[i]]=true;
                ++i;
            }
            if(i>=s){
                vector<int> r(s);
                for(int j=0;j<s;++j)
                    r[j]=nums[states[j]];
                result.push_back(r);
                --i;
            }
            else
                states[i]=0;
            else
                --i;
        }
        return result;
    }
};

```

👍 获赞 1    💬 添加评论

**Jet**

更新于 6/9/2020, 7:03:51 AM

Permutation which has the repeat element.

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution {
public:
    /**
     * @param : A list of integers
     * @return: A list of unique permutations
     */
    vector<vector<int>> permuteUnique(vector<int> &nums) {
        // write your code here

        vector<vector<int>> result;
        vector<int> per;
        vector<bool> visited(nums.size(), false);

        sort(nums.begin(), nums.end());
        dfs(nums, result, per, visited);
        return result;
    }

    void dfs(vector<int>& nums, vector<vector<int>>& result, vector<int>& per, vector<bool>& visited) {
        if(per.size() == nums.size()) {
            result.push_back(per);
            return;
        }

        for(int i = 0; i < nums.size(); i++) {
            if(visited[i]) {
                continue;
            }
            if(i > 0 && nums[i] == nums[i-1]) {
                if(visited[i-1] == false) {
                    continue;
                }
            }

            visited[i] = true;
            per.push_back(nums[i]);
            dfs(nums, result, per, visited);
            visited[i] = false;
            per.pop_back();
        }
    }
};
```

👍 获赞 1

💬 添加评论

**九章用户SBIW5K**

更新于 6/9/2020, 7:04:27 AM

之前答案的代码在递归的出口缺一个判断results中是否已经存在subsets的判断语句

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ /
Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
class Solution {
    /**
     * @param nums: A list of integers.
     * @return: A list of unique permutations.
     */
    public List<List<Integer>> permuteUnique(int[] nums) {
        // Write your code here
        List<List<Integer>> result = new ArrayList<>();
        List<Integer> subset = new ArrayList<>();
        if (nums == null){
            return result;
        }

        if (nums.length == 0){
            result.add(new ArrayList<Integer>());
            return result;
        }

        int[] visited = new int[nums.length];
        for (int i = 0; i < visited.length; i++){
            visited[i] = 0;
        }

        helper(nums, result, subset, visited);

        return result;
    }

    private void helper(int[] nums, List<List<Integer>> result, List<Integer> subset, int[] visited){
        if (subset.size() == nums.length && !result.contains(subset)){
            result.add(new ArrayList<Integer>(subset));
            return;
        }

        for (int i = 0; i < nums.length; i++){
            if (visited[i] == 1 || (i != 0 && nums[i] == nums[i - 1] && visited[i - 1] == 0)){
                continue;
            }
            visited[i] = 1;
            subset.add(nums[i]);
            helper(nums, result, subset, visited);
            subset.remove(subset.size() - 1);
            visited[i] = 0;
        }
    }
}
```

👍 获赞 0

💬 添加评论

加载更多题解

## 进阶课程

直播+互动

直播+互动

直播+互动

互动课

### 九章算法班 2021 版

8周时间精通 57 个核心高频考点, 9 招击破 FLAG、BATJ 算法面试。22....

### 系统架构设计 System Design 2021 版

成为百万架构师必上。30 课时带你快速掌握 18 大系统架构设计知识点与面...

### 九章算法面试高频题冲刺班

每期更新 15% 题目, 考前押题, 一举拿下 FLAG & BATJ Offer



### 面向对象设计 OOD

应届生及亚马逊面试必考, IT 求职必备基础

[首页 \(/?skip\\_redirect=true\)](#) | [联系我们 \(mailto:info@jiuzhang.com\)](mailto:info@jiuzhang.com) | [加入我们 \(/joinus\)](#)

Copyright © 2013-2021 九章算法 浙ICP备19045946号-1  
(<http://www.miibeian.gov.cn/>)

商务合作: [fukesu@jiuzhang.com \(mailto:fukesu@jiuzhang.com\)](mailto:fukesu@jiuzhang.com)

 (<http://weibo.com/ninechapter>)  知  
(<https://www.zhihu.com/people/crackinterview/>)

(/)