

```
1  /*
2  给定一个没有重复数字的序列，返回其所有可能的全排列。
3
4  示例：
5
6  输入：[1,2,3]
7  输出：
8  [
9      [1,2,3],
10     [1,3,2],
11     [2,1,3],
12     [2,3,1],
13     [3,1,2],
14     [3,2,1]
15 ]
16
17 来源：力扣（LeetCode）
18 链接：https://leetcode-cn.com/problems/permutations
19 著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
20  */
```

分析:

- 递归法:依次递归每一种排列可能,直到数组长度为1,即可推入结果容器.

方法一:C++_递归法

```
1  class Solution
2  {
3      private:
4          vector<vector<int>> ret_val;
5          void __rank(vector<int>& nums,vector<int> level_rank)
6          {
7              if(nums.size() == 1)
8              {
9                  level_rank.push_back(nums[0]);
10                 ret_val.push_back(level_rank);
11                 return ;
12             }
13             for(int i = 0; i < nums.size() ; i++)
14             {
15                 vector<int> temp;
16                 for(int j = 0 ; j < nums.size() ; j++)
17                 {
18                     if(i!=j)
19                     {
20                         temp.push_back(nums[j]);
21                     }
22                     else
23                     {
24                         level_rank.push_back(nums[j]);
25                     }
26                 }
27             }
28         }
29     };
30 }
```

```

26         }
27
28         __rank(temp, level_rank);
29         level_rank.pop_back();
30     }
31 }
32
33 public:
34     vector<vector<int>> permute(vector<int>& nums)
35     {
36         ret_val.clear();
37         vector<int> temp;
38         if(nums.size() !=0)
39         {
40             __rank(nums,temp);
41         }
42         return ret_val;
43     }
44 };
45
46 /*
47 执行结果:
48 通过
49 显示详情
50 执行用时 :20 ms, 在所有 C++ 提交中击败了63.44% 的用户
51 内存消耗 :10.4 MB, 在所有 C++ 提交中击败了10.14%的用户
52 */

```