

全排列

全排列问题的算法是非常简单且典型的回溯算法的用例。

力扣

<https://leetcode-cn.com/problems/permutations/submissions/>

```
class Solution {
public:
    vector<vector<int>> permute(vector<int>& nums) {
        vector<vector<int>> answer;
        // 回溯算法是树形结构算法之一，树形算法使用递归是很理所当然的事情
        recursivePermute(nums, 0, answer);
        return answer;
    }

    void recursivePermute(vector<int>& nums, int begin, vector<vector<int>>& answer) {
        if (begin >= nums.size()) {
            answer.push_back(nums);
            return;
        }
        for (int i = begin; i < nums.size(); ++i) {
            swap(nums[begin], nums[i]);
            recursivePermute(nums, begin + 1, answer);
            // 回溯算法的核心（与其他树形算法不同的地方），将尝试过的方案回退。
            swap(nums[begin], nums[i]);
        }
    }
};
```

此处为了能够将结果输出，用了一个小技巧，使用swap把当前选中的数替换到该轮递归相应的位置上。

上述代码很好的诠释了覃超所描述的递归算法的模板，

// terminator

// process current level

// drill down

// revert and clean up