一、题目说明

题目是46. Permutations，给一组各不相同的数，求其所有的排列组合。难度是Medium！

二、我的解答

这个题目，前面遇到过类似的。回溯法（树的深度优先算法），或者根据如下求解：

[刷题31. Next Permutation]

我考虑可以用dp做，写了一个上午，理论我就不说了，自己看代码：

```
#include<iostream>
#include<vector>
#include<unordered_map>

using namespace std;
class Solution{
    public:
        vector<vector<int>> permute(vector<int>& nums) {
            vector<vector<int>> res;
            vector<vector<int>> next;

            unordered_map<int,vector<vector<int>>> dp;
            vector<int> cur;

            if(nums.empty()) return res;

            cur.push_back(nums[0]);
            res.push_back(cur);
            dp[1] = res;
            int currLength = 2;
            for(int j=1;j<nums.size();j++){
                res = dp[j];
                next.clear();

                for(int k=0;k<currLength;k++){
                    cur.clear();
                    cur.resize(j+1);

                    for(int m=0;m<res.size();m++){
                        cur[k] = nums[j];
                        int t1=0,t2=0;
                        while(t2<res[m].size()){
                            if(cur[t1]!=nums[j]){
                                cur[t1] = res[m][t2];
                            }else{
                                ++t1;
                                cur[t1] = res[m][t2];
                            }
                            t1++;
                            t2++;
                        }

                        next.push_back(cur);
                        cur.clear();
                        cur.resize(j+1);
                    }
                }

                currLength++;
                dp[j+1] = next;
            }
            return dp[nums.size()];
        }
};
int main(){
    Solution s;
    vector<int> nums = {1,2,3,4};
    vector<vector<int>> r = s.permute(nums);
    for(int i=0;i<r.size();i++){
        for(int j=0;j<r[i].size();j++){
            cout<<r[i][j]<<" ";
        }
        cout<<"\n";
    }


    return 0;
}
```

性能如下：

```
Runtime: 8 ms, faster than 98.85% of C++ online submissions for Permutations.
Memory Usage: 9.5 MB, less than 46.27% of C++ online submissions for
Permutations.
```

三、优化措施

dp算法，是按照空间换时间的，所以时间还可以，空间就差了点。

下面是回溯算法的代码，可读性好多了：

```
class Solution{
    private:
        vector<vector<int>> result;
        vector<int> path;
        vector<bool> used;
    public:
        //枚举每个位置放哪个数
        void dfs(const vector<int>&nums,int pos){
            if(pos == nums.size()){
                result.push_back(path);
                return;
            }
            for(int i=0;i<nums.size();i++){
                if(!used[i]){
                    path.push_back(nums[i]);
                    used[i] = true;
                    dfs(nums,pos+1);
                    used[i] = false;
                    path.pop_back();
                }
            }
        }
        vector<vector<int>> permute(vector<int>& nums) {
            if(nums.empty()){
                return result;
            }

            used.resize(nums.size());
            dfs(nums,0);
            return result;
        }
};
```