1. Permutations

Given a collection of **distinct** integers, return all possible permutations.

**Example:**

Input: [1,2,3]  
Output:  
[  
 [1,2,3],  
 [1,3,2],  
 [2,1,3],  
 [2,3,1],  
 [3,1,2],  
 [3,2,1]  
]

**解**

深度优先搜索即可

class Solution {  
public:  
 vector<vector<int>> permute(vector<int>& nums) {  
 vector<vector<int> > ans;  
 vector<int>tmp;  
 bool flag[nums.size()] = {false};  
 dfs(nums, ans, tmp, flag);  
 return ans;  
 }  
 void dfs(vector<int>& nums, vector<vector<int> >&ans, vector<int> &tmp, bool flag[]){  
 if(tmp.size() == nums.size()){  
 ans.push\_back(tmp);  
 return;  
 }  
 for(int i = 0; i < nums.size(); ++i){  
 if(flag[i] == true)continue;  
 flag[i] = true;  
 tmp.push\_back(nums[i]);  
 dfs(nums, ans, tmp, flag);  
 tmp.pop\_back();  
 flag[i] = false;  
 }  
 }  
};