

给定一个**没有重复**数字的序列，返回其所有可能的全排列。

**示例:**

**输入:** [1,2,3]

**输出:**

```
[
  [1,2,3],
  [1,3,2],
  [2,1,3],
  [2,3,1],
  [3,1,2],
  [3,2,1]
]
```

思路，全排列就是深度遍历，不遍历已经遍历过的节点，当遍历到最深处时，将数组添加进去，即可完成全排列。

**代码**

```
class Solution {
public:
    void dfs(vector<int>& nums, map<int, int> &bj, vector<vector<int>>& result, int n, vector<int> num) {
        if (n >= 0) {
            num.push_back(nums[n]);
        }
        if (num.size() == nums.size()) {
            result.push_back(num);
            return;
        }
        for (int i = 0; i < nums.size(); i++) {
            if (bj[i] == 0) {
                bj[i] = 1;
                dfs(nums, bj, result, i, num);
                bj[i] = 0;
            }
        }
    }

    vector<vector<int>> permute(vector<int>& nums) {
        vector<int> num;
        vector<vector<int>> result;
        map<int, int> bj;
        dfs(nums, bj, result, -1, num);
        return result;
    }
};
```