

给定一组**不含重复元素**的整数数组 *nums*，返回该数组所有可能的子集（幂集）。

说明：解集不能包含重复的子集。

示例：

输入： *nums* = [1,2,3]

输出：

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

思路，递归遍历数组，数组有两个状态，添加当前元素，不添加当前元素，以这个为分支，把数组添加变成两种情况继续遍历，然后动态的添加遍历的数组，即可得到答案。

代码

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