

一、题目说明

题目136. Single Number，给定一列非空整数，所有数字都出现2次，只有1个数字只出现1次，找到这个数字。

二、我的解答

这个题目，最简单的做法是排序，然后查找即可。在此，我们用 `unordered_map<int,int> dp`; 统计数量，然后找到数量是1的。

```
class Solution{
public:
    int singleNumber(vector<int>& nums){
        if(nums.size()<1) return 0;
        if(nums.size()==1) return nums[0];
        unordered_map<int,int> dp;
        for(int i=0;i<nums.size();i++){
            dp[nums[i]] ++;
        }
        for(int i=0;i<nums.size();i++){
            if(dp[nums[i]]<2) return nums[i];
        }
        return 0;
    }
};
```

性能如下：

```
Runtime: 24 ms, faster than 27.29% of C++ online submissions for Single Number.
Memory Usage: 11.6 MB, less than 20.99% of C++ online submissions for Single Number.
```

三、优化措施

上面是遍历2次，可以优化为遍历1次：

```
class Solution{
public:
    int singleNumber(vector<int>& nums){
        if(nums.size()<1) return 0;
        if(nums.size()==1) return nums[0];
        unordered_map<int,int> dp;
        for(int i=0;i<nums.size();i++){
            dp[nums[i]] ++;
            if(dp[nums[i]]==2) dp.erase(nums[i]);
        }
        unordered_map<int,int>::iterator iter = dp.begin();
        return iter->first;
    }
};
```

不使用额外的空间：

```
class Solution{
public:
    //异或操作
    int singleNumber(vector<int>& nums){
        int n = 0;
        for(int i=0;i<nums.size();i++){
            n = n xor nums[i];
        }
        return n;
    }
};
```

Runtime: 12 ms, faster than 95.18% of C++ online submissions for Single Number.
Memory Usage: 9.7 MB, less than 95.06% of C++ online submissions for Single Number.