

```
1  /*
2  给定一个整数数组 nums，其中恰好有两个元素只出现一次，其余所有元素均出现两次。 找出只出现一
   次的那两个元素。
3
4  示例：
5
6  输入：[1,2,1,3,2,5]
7  输出：[3,5]
8
9  注意：
10
11     结果输出的顺序并不重要，对于上面的例子， [5, 3] 也是正确答案。
12     你的算法应该具有线性时间复杂度。你能否仅使用常数空间复杂度来实现？
13
14  来源：力扣（LeetCode）
15  链接：https://leetcode-cn.com/problems/single-number-iii
16  著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
17  */
```

分析:

- 第一次遍历针对数组,利用map表进行存储出现的次数,第二次遍历针对map,找出只出现一次的元素.

方法一:C++\_

```
1  class Solution
2  {
3      public:
4          vector<int> singleNumber(vector<int>& nums)
5          {
6              vector<int>      ret_val      ;
7              map<int,int>      mii          ;
8              int               i            ;
9              map<int,int>::iterator iter    ;
10
11              for( i = 0; i < nums.size() ; i++)
12              {
13                  if(mii.count(nums[i])==1)
14                  {
15                      mii[nums[i]] = 2;
16                  }
17                  else
18                  {
19                      mii[nums[i]] = 1;
20                  }
21              }
22
23              iter = mii.begin();
24              while(iter!=mii.end())
25              {
26                  if(iter->second == 1)
```

```
27         {
28             ret_val.push_back(iter->first);
29         }
30         iter++;
31     }
32     return ret_val;
33 }
34 };
35 /*
36 执行结果:
37 通过
38 显示详情
39 执行用时 :24 ms, 在所有 C++ 提交中击败了35.58% 的用户
40 内存消耗 :10.9 MB, 在所有 C++ 提交中击败了5.50%的用户
41 */
```