

```

1  /*
2  给定一个范围在  $1 \leq a[i] \leq n$  (  $n$  = 数组大小 ) 的 整型数组，数组中的元素一些出现了两
   次，另一些只出现一次。
3
4  找到所有在  $[1, n]$  范围之间没有出现在数组中的数字。
5
6  您能在不使用额外空间且时间复杂度为 $O(n)$ 的情况下完成这个任务吗？你可以假定返回的数组不算在额
   外空间内。
7
8  示例：
9
10 输入：
11 [4,3,2,7,8,2,3,1]
12
13 输出：
14 [5,6]
15
16 https://leetcode-cn.com/classic/problems/find-all-numbers-disappeared-in-an-array/description/
17 */

```

分析:

- 方法一:遍历数组,利用map数据结构保存各个数据出现的次数,然后遍历map筛选出没有出现过的数字;
- 方法二: $1 - 1 \leq a[i] \leq n$,遍历数组,根据元素的数值对对应的索引位置给负值(已经为负值了,说明出现了两次就不必重复给负值了),再遍历一次,对应为正数的索引位置就对应了未出现过的数.

方法一:C++_map

```

1  class solution
2  {
3      public:
4          vector<int> findDisappearedNumbers(vector<int>& nums)
5          {
6              map<int,int> mii;
7              vector<int> ret;
8              int i = 0;
9              for(i=0;i<nums.size();i++)
10             {
11                 mii[nums[i]] = 1;
12             }
13
14             for(i=1;i<=nums.size();i++)
15             {
16                 if(mii.count(i)==0)
17                 {
18                     ret.push_back(i);
19                 }
20             }

```

```

21         return ret;
22     }
23 };
24
25 /*
26 执行结果:
27 通过
28 显示详情
29 执行用时 :228 ms, 在所有 cpp 提交中击败了24.36%的用户
30 内存消耗 :27.9 MB, 在所有 cpp 提交中击败了5.18%的用户
31 */

```

方法二:C++_索引筛选法

```

1  class solution
2  {
3      public:
4          vector<int> findDisappearedNumbers(vector<int>& nums)
5          {
6              vector<int> ret;
7              int i = 0;
8              for(i=0;i<nums.size();i++)
9              {
10                 if(nums[abs(nums[i])-1] >0)
11                 {
12                     nums[abs(nums[i])-1] *= -1;
13                 }
14             }
15
16             for(i=0;i<nums.size();i++)
17             {
18                 if(nums[i] > 0)
19                 {
20                     ret.push_back(i+1);
21                 }
22             }
23             return ret;
24         }
25     };
26     /*
27     执行结果:
28     通过
29     显示详情
30     执行用时 :120 ms, 在所有 cpp 提交中击败了90.19% 的用户
31     内存消耗 :14.8 MB, 在所有 cpp 提交中击败了88.05%的用户
32     */

```