

LCR 004. 只出现一次的数字 II

```
1 class Solution {
2 public:
3     int singleNumber(vector<int>& nums) {
4         int ret = 0;
5         for (int i = 0; i < 32; i++) {
6             int sum = 0;
7             for (auto e : nums)
8                 if ((e >> i) & 1) == 1)
9                     sum++;
10                sum %= 3;
11                if (sum == 1)
12                    ret |= (1 << i);
13            }
14        return ret;
15    }
16 };
```

面试题 17.19. 消失的两个数字

```
1 class Solution {
2 public:
3     vector<int> missingTwo(vector<int>& nums) {
4         vector<int> tmp = nums;
5         for(int i = 0 ; i < nums.size()+3; i++)
6         {
7             tmp.push_back(i);
8         }
9         int sum = 0;
10        for(auto e : tmp) sum ^= e;
11        //到这就变成只出现一次数字三
12        int lowbit = (sum & (-sum));
13        vector<int> ans(2);
14        for (int x: tmp) {
15            ans[(x & lowbit) != 0] ^= x; // 分组异或
16        }
17        return ans;
18    }
19 }
20 };
```