在 2.5 亿个整数中找出不重复的整数

题目标签

学习时长：20分钟

题目难度：中等

知识点标签：位图法、分治法、大数据量处理

题目描述

在 2.5 亿个整数中找出不重复的整数。注意：内存不足以容纳这 2.5 亿个整数。

1. 解答思路

方法一：分治法

与前面的题目方法类似，先将 2.5 亿个数划分到多个小文件，用 HashSet/HashMap 找出每个小文件中不重复的整数，再合并每个子结果，即为最终结果。

方法二：位图法

位图，就是用一个或多个 bit 来标记某个元素对应的值，而键就是该元素。采用位作为单位来存储数据，可以大大节省存储空间。

位图通过使用位数组来表示某些元素是否存在。它可以用于快速查找，判重，排序等。不是很清楚？我先举个小例子。

假设我们要对 `[0,7]` 中的 5 个元素 (6, 4, 2, 1, 5) 进行排序，可以采用位图法。0~7 范围总共有 8 个数，只需要 8bit，即 1 个字节。首先将每个位都置 0：

```
0 0 0 0 0 0 0 0
```

然后遍历 5 个元素，首先遇到 6，那么将下标为 6 的位的 0 置为 1；接着遇到 4，把下标为 4 的位 的 0 置为 1：

```
0 0 0 0 1 0 1 0
```

依次遍历，结束后，位数组是这样的：

```
0 1 1 0 1 1 1 0
```

每个为 1 的位，它的下标都表示了一个数：

```
for i in range(8):    if bits[i] == 1:        print(i)
```

这样我们其实就已经实现了排序。

对于整数相关的算法的求解，**位图法**是一种非常实用的算法。假设 int 整数占用 4B，即 32bit，那么我们可以表示的整数的个数为 232。

那么对于这道题，我们用 2 个 bit 来表示各个数字的状态：

•00 表示这个数字没出现过；•01 表示这个数字出现过一次（即为题目所找的不重复整数）；•10 表示这个数字出现了多次。

那么这 232 个整数，总共所需内存为 232*2b=1GB。因此，当可用内存超过 1GB 时，可以采用位图法。假设内存满足位图法需求，进行下面的操作：

遍历 2.5 亿个整数，查看位图中对应的位，如果是 00，则变为 01，如果是 01 则变为 10，如果是 10 则保持不变。遍历结束后，查看位图，把对应位是 01 的整数输出即可。

2. 方法总结

判断数字是否重复的问题，位图法是一种非常高效的方法。