

位图

前置知识：讲解003-二进制和位运算、讲解005-对数器

特别提醒：Python的同学实现位运算的题目需要特别注意，需要自己去手动处理溢出和符号扩展等问题

比如：(n << shift_amount) & 0xFFFFFFFF

位图原理

其实就是用bit组成的数组来存放值，用bit状态1、0代表存在、不存在，取值和存值操作都用位运算
限制是必须为连续范围且不能过大。好处是极大的节省空间，因为1个数字只占用1个bit的空间。

位图的实现

Bitset(int n): 初始化位图的大小，只支持0~n-1所有数字的增删改查

void add(int num): 把num加入到位图

void remove(int num): 把num从位图中删除

void reverse(int num): 如果位图里没有num，就加入；如果位图里有num，就删除

boolean contains(int num): 查询num是否在位图中

将采用对数器验证，当你找不到测试链接的时候就用对数器验证，而且对数器验证更稳妥、更能练习debug能力

位图

找到了一个相关测试: <https://leetcode-cn.com/problems/design-bitset/>

Bitset是一种能以紧凑形式存储位的数据结构

Bitset(int n): 初始化n个位, 所有位都是0

void fix(int i): 将下标i的位上的值更新为1

void unfix(int i): 将下标i的位上的值更新为0

void flip(): 翻转所有位的值

boolean all(): 是否所有位都是1

boolean one(): 是否至少有一位是1

int count(): 返回所有位中1的数量

String toString(): 返回所有位的状态

位图的后续扩展, 将在【扩展】课程里进一步讲述