“1 与 0 ，
一切数字的神奇渊源。
这是造物的秘密美妙的典范，
因为，
一切无非都来自上帝。”

浅谈信息学竞赛中的“0”和“1”
—二进制思想在信息学竞赛中的应用

河北省石家庄二中
武森

content

二进制思想在数据结构中的应用

- 树状数组

二进制思想在解题思想中的应用

- 状态压缩
- 模型转化
- 01 二叉树

例题一： Matrix

- 有一个 M*N 的矩阵，每一个格子中的数是 1 或 0 ，初始时为 0 。有两种操作：
  - 修改一个子矩阵，将子矩阵中的数字全部 01 取反。
  - 查询第 x 行第 y 列的格子中的数字 。

退而求其次

如果给定的是一个长度为 N 的一排格子。

每次可以修改一个子列中的数字。

根据这个题目中介绍的这个矩阵中的
数的特点不是 1 就是 0 ，这样我们只需记录
每个格子改变过几次，即可判断这个格子的数字。

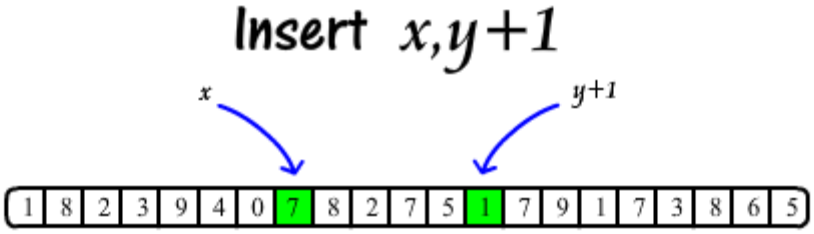
这样，这道题目就变得简单了！

修改：

每次修改的时候，不妨把格子修改的范围 (x,y) 变成两个点，一个为更改的初始节点
x ，另一个为更改的终止节点 y+1 ，然后往这列格子中的这两个节点中加 1。

修改：

每次修改的时候，不妨把格子修改的范围 (x,y) 变成两个点，一个为更改的初始节点
x ，另一个为更改的终止节点 y+1 ，然后往这列格子中的这两个节点中加 1。

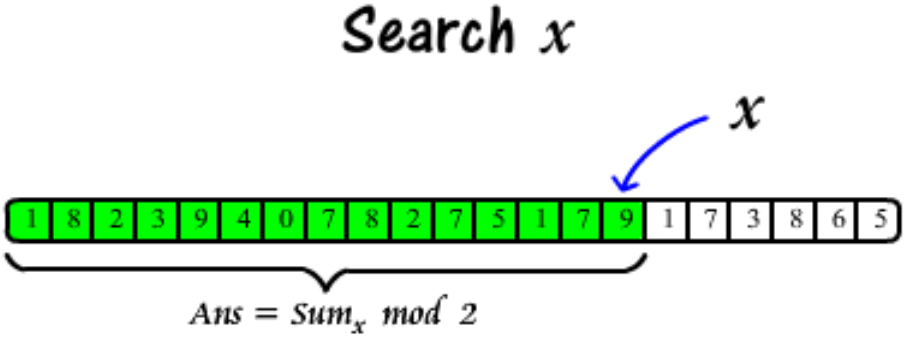


查询

每次询问的时候只需计算出 Sumx 就可以求出第 x 个格子被修改过几次。

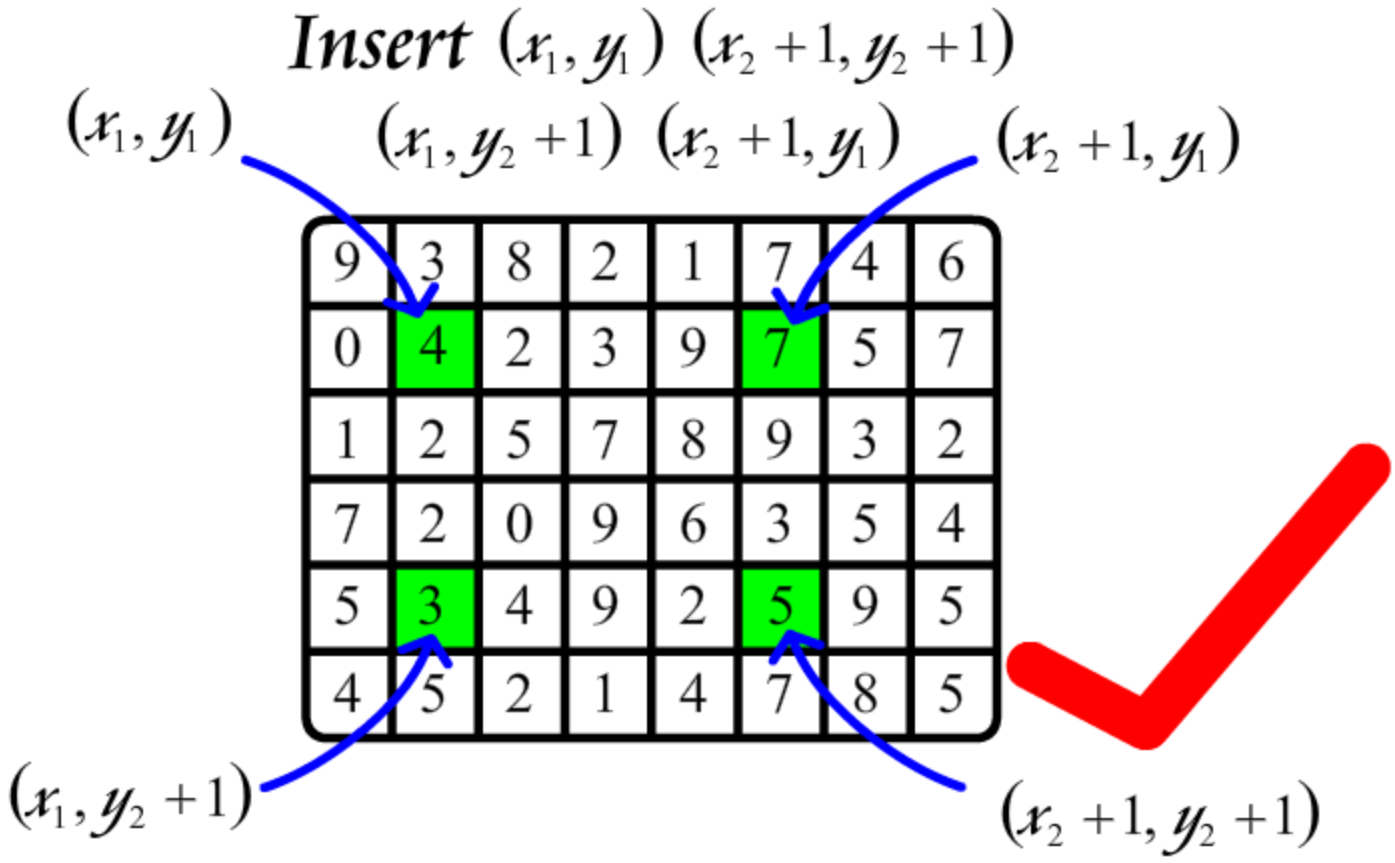
查询

每次询问的时候只需计算出 Sumx 就可以求出第 x 个格子被修改过几次。

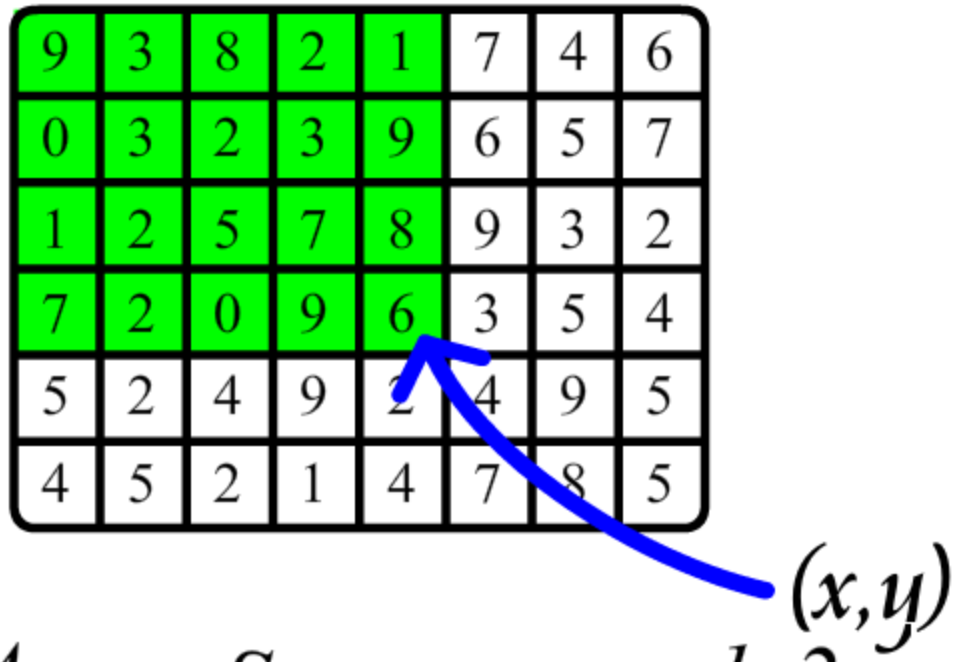


寻根溯源

用上面的方法看看能否解决原来的问题。


Insert (x_1, y_1) (x_2+1, y_2+1)
(x_1, y_1)
(x_1, y_2+1)
(x_2+1, y_1)
(x_2+1, y_1)
9 3 8 2 1 7 4 6
0 4 2 3 9 7 5 7
1 2 5 7 8 9 3 2
7 2 0 9 6 3 5 4
5 3 4 9 2 5 9 5
4 5 2 1 4 7 8 5
(x_1, y_2+1)
(x_2+1, y_2+1)

Search x,y



$$Ans = Sum_{x,y} \mod 2$$

推而广之

如果是要处理三维的情况，甚至 N 维的情况呢？

一般的数据结构能解决吗？

不能！！

怎么办呢？？？

二进制思想

二进制思想在数据结构中的应用：

树状数组中的每一个元素的编号变成了二制编码，再通过这些二进制编码末尾的 0 的个数来决定存储什么信息，假设节点编号为 x ，那么这个节点存储数据的区间为 2^k（其中 k 为 x 二进制末尾 0 的个数）个元素。

又由于每个十进制数转化成二进制位的话，1 的个数最多只有 O(logN) 个，所以，复杂度只有 O(logN) 。

具体操作：

2^k ： X and –X

插入或删除：

```
While x<=max do
Begin
C[x]:=c[x]+delta;
X:=x+(x and –x);
End;
```

查询：

```
While x>0 do
Begin
Sum:=sum+c[x];
X:=x-(x and –x);
End;
```

树状数组

优势

代码长度短，不易出错。

思想巧妙，算法复杂度低。

维护简单，空间消耗低。

易推广到二维甚至三维等等。

例题二： Cow Xor

农民约翰在喂奶牛的时候被另一个问题卡住了。他的所有 N(1 <=N <= 100,000) 个奶牛在他面前排成一行 (按序号 1..N 的顺序) ，按照它们的社会等级排序。奶牛 #1 由最高的社会等级，奶牛 #N 最低。每个奶牛同时被赋予了一个唯一的数在 $0..2^{21}-1$ 的范围内。帮助农民约翰找出应该从那一头奶牛开始喂，使得从它开始的某一个连续的子序列上的奶牛的数的异或值最大。如果有多个这样的子序列，选择结尾的奶牛社会等级最高的。如果还不唯一，选择最短的。

根据异或的性质，可以得出以下结论：

$Sum_k = a_1 \text{ xor } a_2 \text{ xor } a_3 \ldots a_{k-1} \text{ xor } a_k$

$a_i \text{ xor } a_{i+1} \ldots a_{j-1} \text{ xor } a_j = Sum_j \text{ xor } Sum_{i-1}$

- 直接枚举起始点和终结点？

时间复杂度是 **O(N*N)**

Impossible!!!

二进制思想

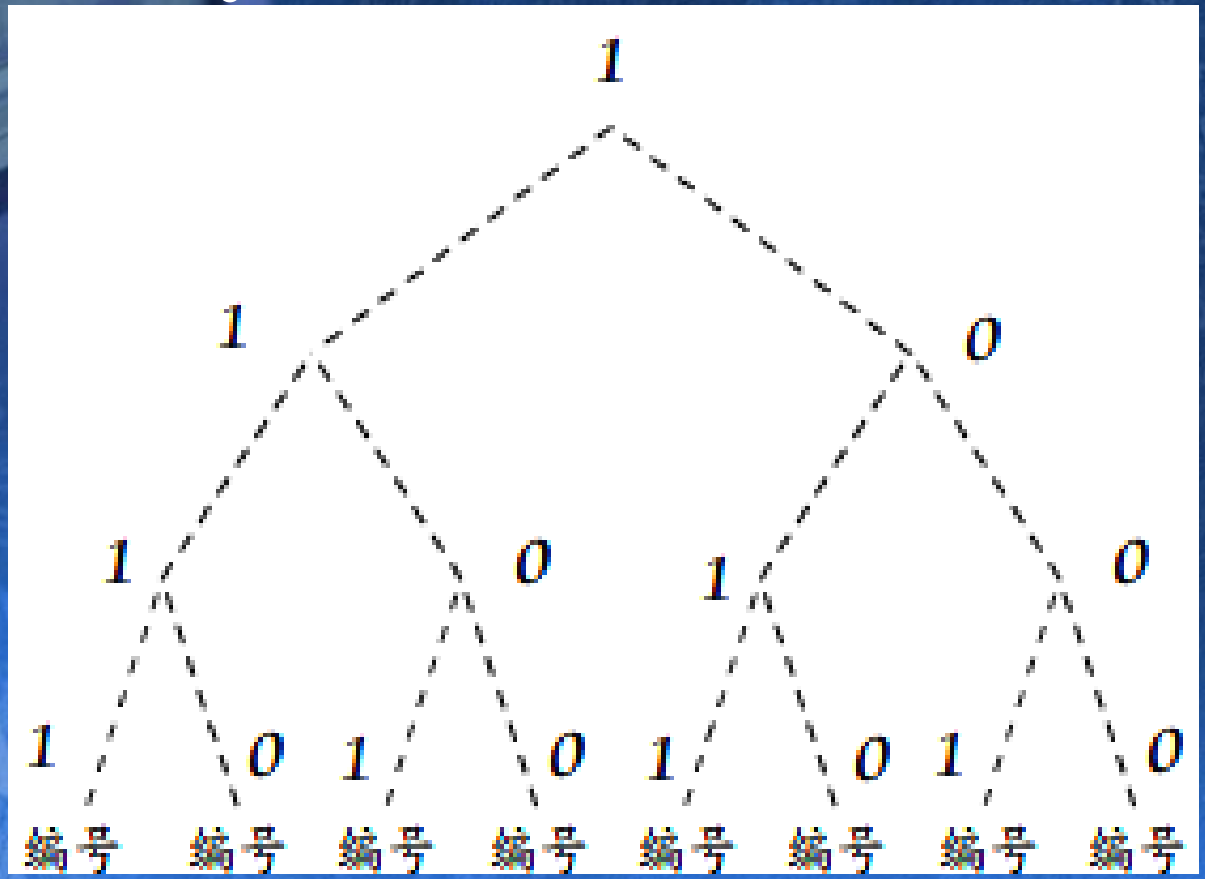
- 数的范围在 $0..2^{21} - 1$ 的整数
- 把这些数转化成二进制只有 21 位

有用吗？？？

Of Course！！！

01 二叉树

顾名思义，树的节点的值为 0 或 1。

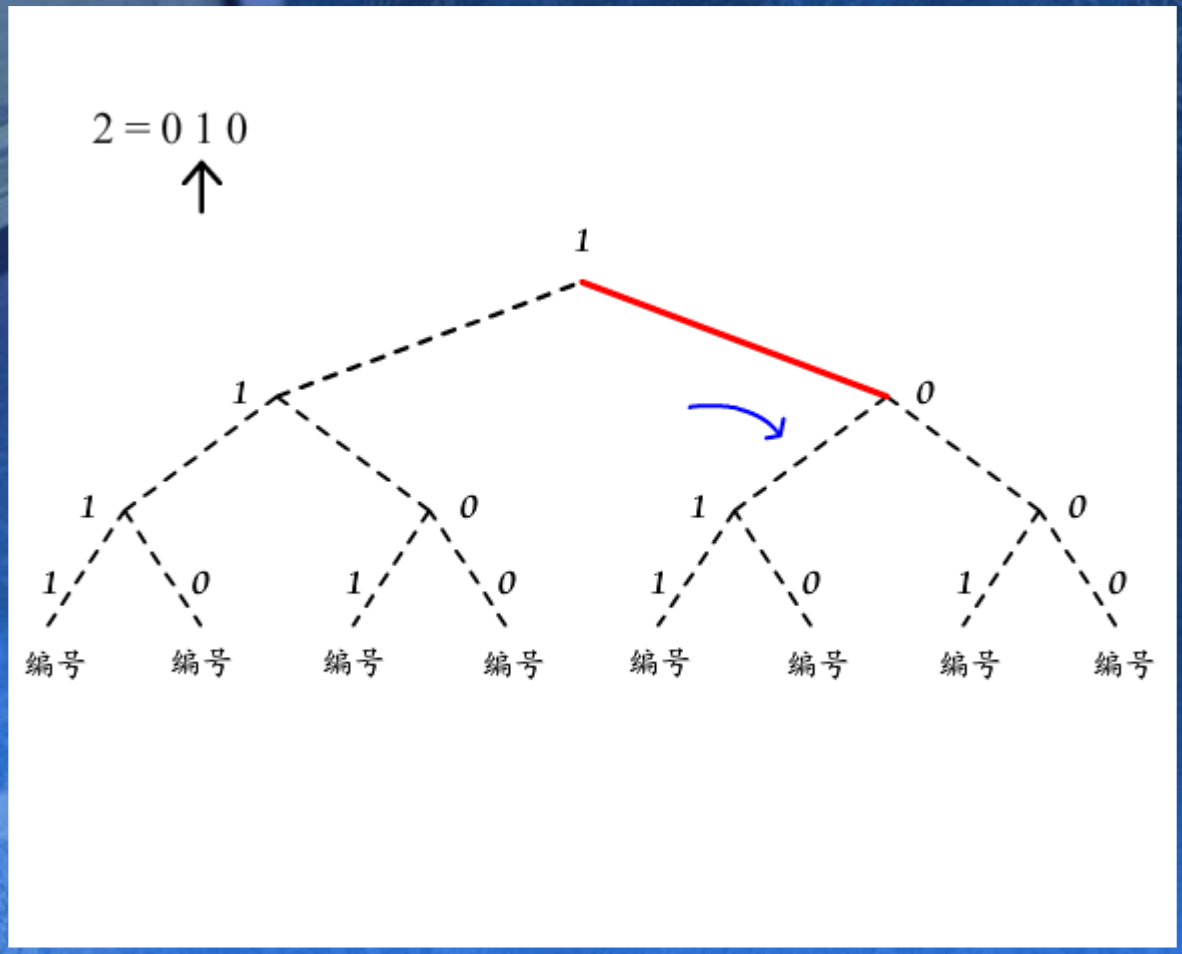


插入

- 每次插入的时候，根据这个数的二进
- 制数进行建树，第 i 位是 1 则向左儿子
- 建一条边，反之向右儿子建边。

插入

- 每次插入的时候，根据这个数的二进
- 制数进行建树，第 i 位是 1 则向左儿子
- 建一条边，反之向右儿子建边。

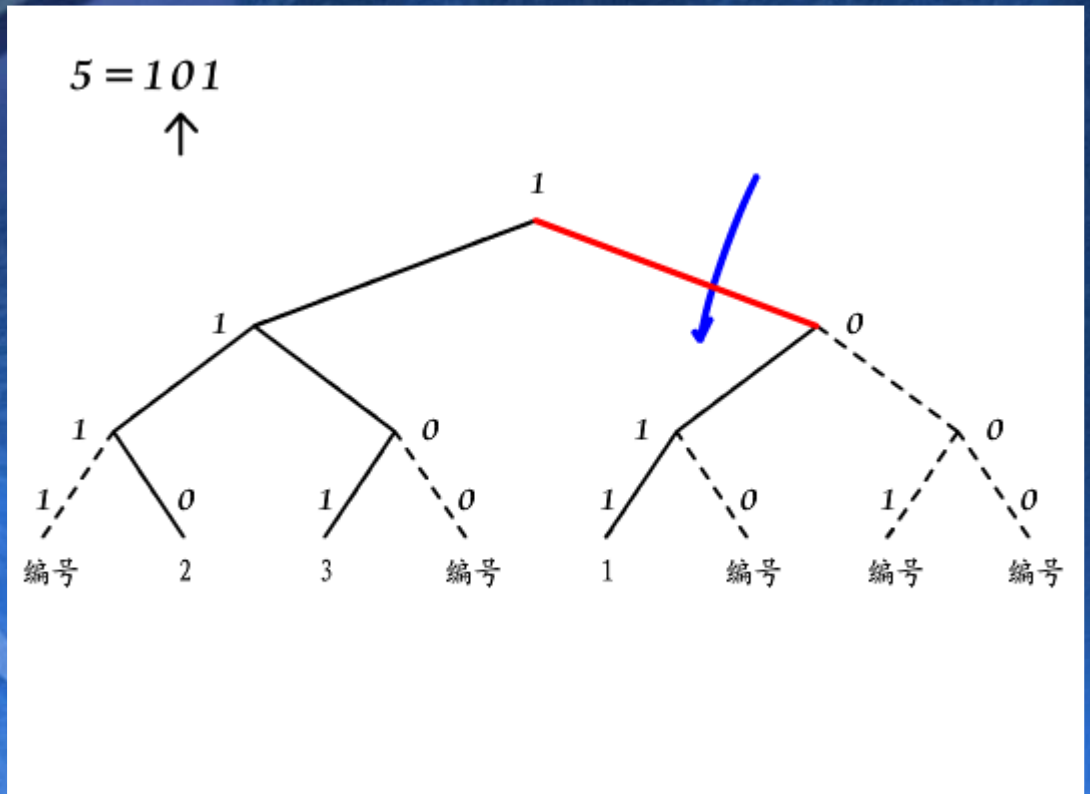


查询

- 每次查询的时候，用贪心的思想根
- 据这个数的二进制数进行，第 i 位是
- 1 如果有右儿子则向右儿子进行，反
- 之向左儿子进行。

查询

- 每次查询的时候，用贪心的思想根
- 据这个数的二进制数进行，第 i 位是
- 1 如果有右儿子则向右儿子进行，反
- 之向左儿子进行。



这样，每次插入和查询的时间复杂度为 **O(logN)** 的，对与这道题目整体的时间复杂度为 **O(NlogN)** 。

这道题目完美解决 ~

总结

二进制思想在信息学竞赛中的应用，巧妙的运用了十进制数与二进制数之间的关系，不仅在数据结构中有广泛应用。

在解题中，将二进制思想引入，不仅可以用于状态压缩，还可以用与构建新的数学模型。

从而达到**转十为二，事半功倍**的效果！


Thank You ！
欢迎提问 ~
WuSen