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

前言

在德国图灵根著名的郭塔王宫图书馆（Schlossbiliothke zu Gotha）保存着一份弥足珍贵的手稿，其标题为：“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。”

众所周知，二进制是计算技术中广泛采用的一种数制,现代的电子计算机技术全部采用的是二进制，因为它只使用 0、1 两个数字符号，非常简单方便，易于用电子方式实现。计算机内部处理的信息，都是采用二进制数来表示的。二进制（Binary）数用 0 和 1 两个数字及其组合来表示任何数。进位规则是“逢 2 进 1”，数字 1 在不同的位上代表不同的值，按从右至左的次序，这个值以二倍递增。除了数值外，英文字母、符号、汉字、声音、图象等数据在计算机内部也采用二进制数的形式来编码。目前最常用的是使用国际标准代码 ASCII 码（美国标准信息交换码）。二进制思想在信息学竞赛中也有广泛的应用。本文通过几个例子，总结了二进制思想在信息学竞赛中的应用。

关键字　二进制 十进制 树状数组 状态压缩 01 二叉树

正文

第一章：二进制思想在数据结构中的应用

例题一：Matrix

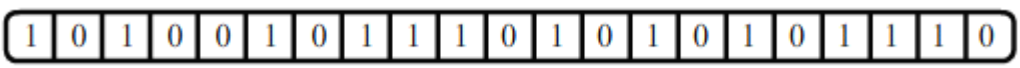
提供一个 M*N 的矩阵，其中每一个格子中的数不是 1 就是 0，初始时每一个格子的值为 0，我们可以修改这个矩阵中的数字，每次给出矩阵的左上角坐标(x_1, y_1)，以及右下角的坐标(x_2, y_2)，并且将矩阵中的数字全部取反（原来是 1 现在变成 0，原来是 0 现在变成 1），还可以每次查询第 x 行第 y 列的格子中的数字是什么。

分析：

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。

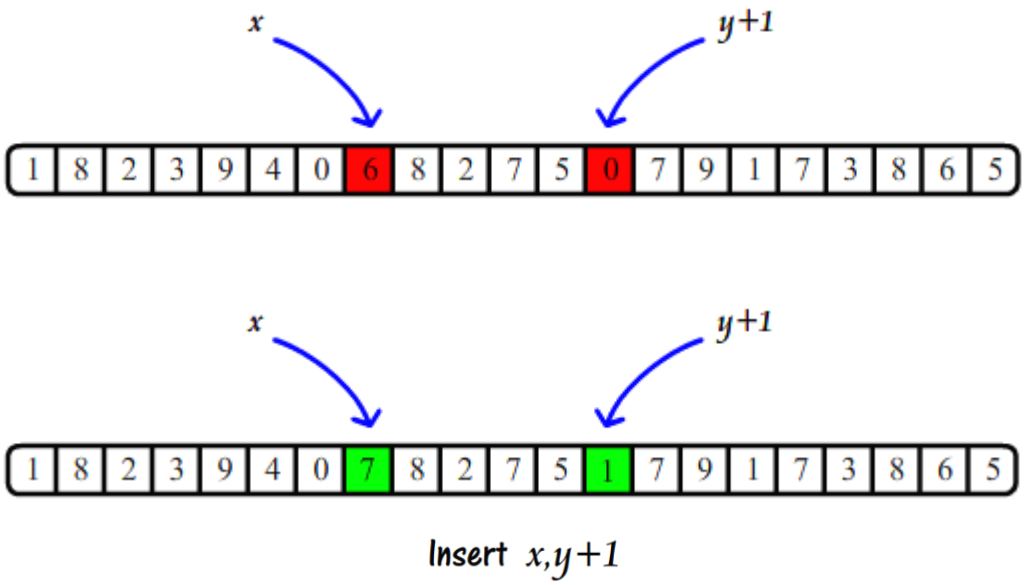
- **第一步**

我们不妨把这道题目简化一下，假定题目中给定的是长度为 N 的一列格子 d。

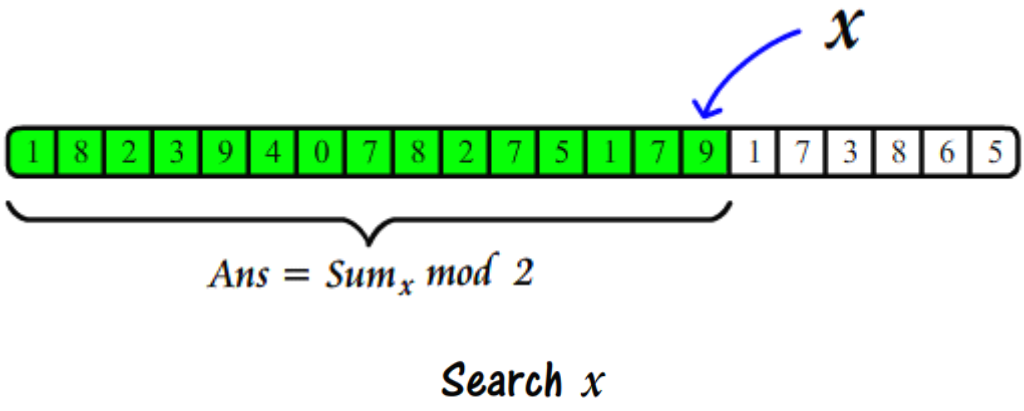


这样，这道题目就变得简单了。

每次修改的时候，给定一个格子修改的范围(x,y)，这样我们不妨把这个范围变成两个点，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。



每次询问x的时候只需计算出$Sum_x=\sum_{i=1}^{x}d_i$这样就可以求出第x个格子被修改过几次，输出的答案就是$Sum_x \mod 2$。



通过以上的方法，我们用一般的数据结构就可使得插入的复杂度

为 $O(\log N)$，查询的复杂度为 $O(\log N)$。

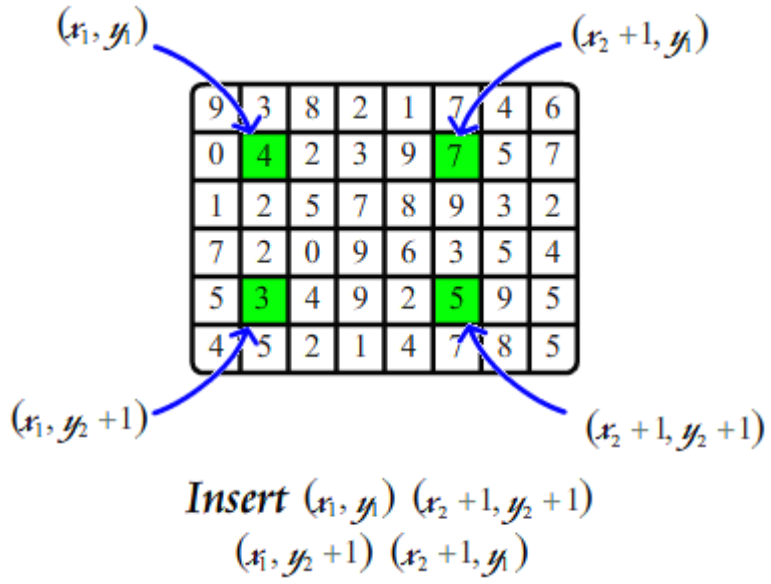
这样一维的问题我们就完美的解决了！

● 第二步

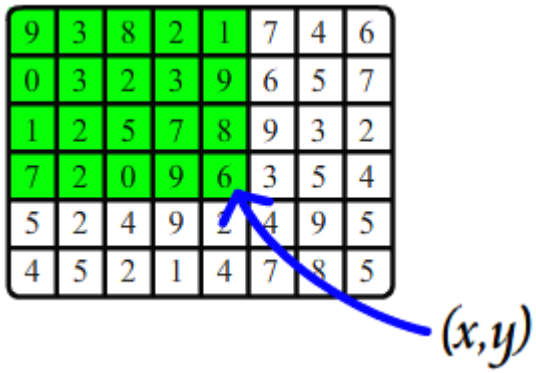
我们已经解决了一维的问题，接下来我们就可以看看题目中的二维情况。

我们能不能用上面的方法解决这一道题目呢？

通过分析我们只改变两个格子的数字保证不了要求的性质(只改变矩阵中的数字而不改变其他的数字)，由于一维的时候，我们加的两个点实际上给改变的区间定了一个范围，那么二维的情况，我们也给它设定一个范围，加上四个格子 $(x_1,y_1),(x_2+1,y_1),(x_1,y_2+1),(x_2+1,y_2+1)$ 每次插入的时候往这四个格子中加 1。



查询 (x,y) 的时候输出 $Sum_{x,y}=\sum_{i=0,j=0}^{i=x,j=y}d_{i,j}$ 即可。



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

Search x,y

这样做是否正确呢？

证明：

假设：插入 $(x_1,y_1),(x_2+1,y_1),(x_1,y_2+1),(x_2+1,y_2+1)$ 四个值，查询 (a,b)。不妨分类讨论：

如上图所示。

当 (a,b) 属于第 1、2、3、4 或 7 这五个区域时，计算 $Sum_{a,b}$ 不受插入的影响；

当 (a,b) 属于第 5 个区域时，$Sum_{a,b}$ 会受到 (x_1,y_1) 的影响，$Sum_{a,b}$ 相比以前会增加 1，这个更改是正确的。

当 (a,b) 属于第 6 个区域时，$Sum_{a,b}$ 会受到 $(x_1,y_1),(x_2+1,y_1)$ 的影响，

$Sum_{a,b}$相比以前会增加 2，答案是 $Sum_{a,b}$ mod 2，结果不受影响，这个更改是正确的。

当(a,b)属于第 8 个区域时，$Sum_{a,b}$会受到$(x_1,y_1),(x_1,y_2+1)$的影响，$Sum_{a,b}$相比以前会增加 2，答案是 $Sum_{a,b}$ mod 2，结果不受影响，这个更改是正确的。

当 (a,b) 属于第 9 个区域时，$Sum_{a,b}$ 会受到$(x_1,y_1),(x_2+1,y_1),(x_1,y_2+1),(x_2+1,y_2+1)$的影响，$Sum_{a,b}$相比以前会增加 4，答案是 $Sum_{a,b}$ mod 2，结果不受影响，这个更改是正确的。

证明完毕。

通过证明我们发现以上的方法是正确的。

- **第三步**

那么二维的可以解决，三维的呢？N 维的呢？

根据上面的方法，我们不难想到，如果是三维的话，应该在长方体的周围加入 8 个点，N 维的情况，应该在 N 维图形周围加入2^n个点来处理这些情况。统计 $Sum_{i_1,i_2\ldots,i_n}$即可。

- **第四步**

这道题的方法我们已经很明确了，要用到数据结构来解决，但是用线段树等数据结构的话，如果推广到二维或者三维，可能写起来就相当复杂，并且出错的概率相当大，那么有没有一个写起来既简单快捷又易推广的数据结构呢？树状数组!!!

树状数组就是二进制思想的经典应用。

树状数组中的每一个元素的编号变成了二进制编码，如：$11=(1011)_2$，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。因为这个区间最后一个元素必然为 $A[x]$, 这个区间存储的数据为 $A[n-2^k+1]+...+A[n]$。

算出 2^k 可以直接运用位运算：

```
2^k：X and –X
```

插入或删除操作就可以写成

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

查询操作就可以写成

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

这样，插入、删除和查询的最坏时间复杂度为 O(logN)，丝毫不逊色与其它数据结构。

证明：

X-（x and -x）这一步实际上等价于将 x 的二进制的最后一个 1 减去。而 x 的二进制里最多有 log(n)个 1，所以查询效率是 log(n)。

至于修改，修改一个节点，必须修改其所有祖先，最坏情况下为修改第一个元素，最多有 log(n)的祖先。所以修改效率是 log(n)。

证明完毕。

总结：

在数据结构中的运用二进制思想，创造出了一种新的数据结构——树状数组。其思想核心在于运用了十进制数与二进制数之间的联系，通过数的二进制形式来决定储存信息，把复杂的问题简单化,方法简单巧妙。

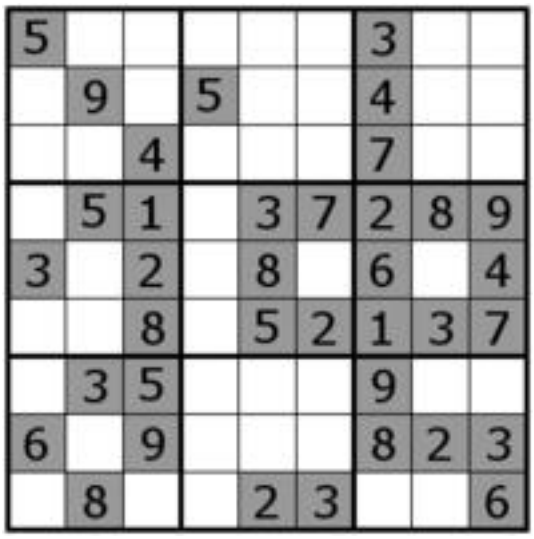
树状数组的优势在于代码长度短，不易出错，思想巧妙，算法复杂度低，维护简单，易推广到二维甚至三维等等。

对于数据结构要求操作不复杂的题目，是上佳的选择。

第二章：二进制思想在解题思路中的应用

例题二：Sudoku

数独盘面是个九宫，每一宫又分为九个小格。在这八十一格中给出一定的已知数字，利用逻辑和推理，在其他的空格上填入 1-9 的数字。使 1-9 每个数字在每一行、每一列和每一宫中都只出现一次。



分析：

- **第一步**

这是一道经典的数独题目，通过给定的数字进行分析和排除，算出其他格子的数字。一种比较容易想到的做法，按照顺序枚举格子中的数，进行搜索，直到搜完为止。

但是数独中的数字排列千变万化，那么究竟有多少种终盘的数字组合呢？6,670,903,752,021,072,936,960（约有 6.67×10 的 21 次方）种组合，单凭这样简单的搜索是不可能完成的。

- **第二步**

我们需要非常有效的剪枝来提高搜索效率。

把每一个空格子可能放的数字记录到表格中，把可能性唯一的数字进行填充，然后按可能放的数的个数进行排序，按可能的个数从小到大进行搜索，每次搜索到一个格子的时候，随机选择的一个可以放得数字填到空格中，并继续进行搜索，但是在实现起来相当困难，每次填一个数字后，该格子所在的行、列以及 3*3 的格子，可以填的数字个数都得修改，修改完了还需要排序，并且写完之后，结果发现还是 TLE!!!

- **第三步**

应用二进制思想，把状态进行状态压缩，将每一个格子想象成一个 9 位的二进制数，使第 1 位…第 9 位，分别表示数字 1…9 是否能放，每个数位上用 0 或 1 来表示，0 表示这个数字可以放，1 表示这个数字不能放。这样就把每个格子表示成 0~511 中的一个数，这样每次搜索的时候，就直接枚举一个数字，通过位运算计算出这行、列以及 3*3 的块中是否可放即可，

通过这样的状态压缩，不用其它的剪枝就可以解决这道题目了。

当然再加上按可能放的数的个数进行排序，按可能的个数从小到大进行搜索之类的优化可以很完美的解决这道题目！！！

例题三：Requirements

给定 N(1 <= N <= 100000)个五维的点 $A(x_1,x_2,x_3,x_4,x_5)$，求两个点 $X(x_1,x_2,x_3,x_4,x_5)$和 $Y(y_1,y_2,y_3,y_4,y_5)$，使得他们的哈密顿距离（即 $|x_1-y_1|+|x_2-y_2|+|x_3-y_3|+|x_4-y_4|+|x_5-y_5|$）最大。

分析：

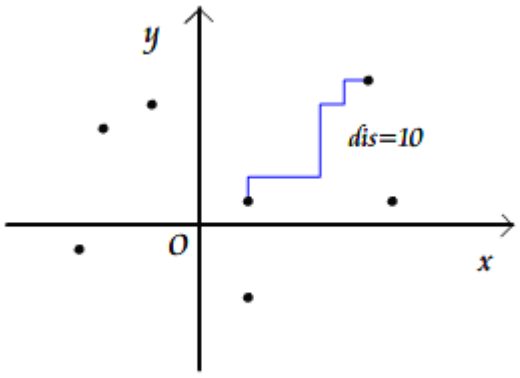
- **第一步**

显然，暴力枚举的 O(N²)的算法会超时，那么怎么办呢？通过读题，我们发现维数远远小于点数，这个信息有用吗？

我们不妨先分析一下这个问题的退化版本。

我们先来看看给定的是 N 个一维点，那么算法很明显，只需扫描一边，记录下最大值、最小值即可得出答案。但是，我们还没有看到这道题目的本质。

我们来分析一下如果是 N 个二维的点，那么我们可以怎么用较快的方法求出 $\max\left(|x_i-y_i|+|x_j-y_j|\right)$的解呢？



通过简单的数学变形，我们可以得到这样的数学公式：

$$|x_i-y_i|+|x_j-y_j| = \max\begin{cases}(x_i-y_i)+(x_j-y_j),\\(x_i-y_i)-(x_j-y_j),\\-(x_i-y_i)+(x_j-y_j),\\-(x_i-y_i)-(x_j-y_j)\end{cases}$$
$$= \max\begin{cases}x_i-y_i+x_j-y_j,\\x_i-y_i-x_j+y_j,\\-x_i+y_i+x_j-y_j,\\-x_i+y_i-x_j+y_j\end{cases}$$

通过观察，我们发现每一对相同元的符号必定相反，如：x_i-y_i，于是我们有了一个二进制思想的思路，那就是枚举这些二维的点的 x 轴 y 轴前的正负号，这样就可以用一个 0~3 的数的二进制形式来表示每个元素前面的正负号，1表示+号,0表示−号，如：2 表示的二进制位形式为 10_2 表示 x_i-x_j。这样我们就可以通过 2^2*N 次记录下这些二元组的不同的符号的数值，对于每个二进制来表示的不同的式子只需记录下他们的 $\max_i$ 和 $\min_i$ 值，这样我们只需求出这些相同的二进制表示的式子 $\max_i-\min_i$，最后我们就可以解出这个问题的解：

$$Ans = \max\begin{cases}\max_0-\min_0\\\max_1-\min_1\\\max_2-\min_2\end{cases}$$

但是这个解对吗？？？

证明：

首先，我们要证明如下公式的正确性，

$$|x_i-y_i|+|x_j-y_j| = \max\begin{cases}(x_i-y_i)+(x_j-y_j),\\(x_i-y_i)-(x_j-y_j),\\-(x_i-y_i)+(x_j-y_j),\\-(x_i-y_i)-(x_j-y_j)\end{cases}$$
$$= \max\begin{cases}x_i-y_i+x_j-y_j,\\x_i-y_i-x_j+y_j,\\-x_i+y_i+x_j-y_j,\\-x_i+y_i-x_j+y_j\end{cases}$$

设 $a=x_i-y_i$， $b=x_j-y_j$。

若 $a\times b=0$

不妨设 $a=0$.

则 $|a|+|b|=|b|=\pm b$

当 $b\geq 0$ 时， $|b|=b\geq -b$ 成立。

当 $b<0$ 时， i 成立。

若 $a\times b>0$

不妨设 $a>0,b>0$且$a\geq b$

则 $|a|+|b|=a+b>a-b>-a+b>-a-b$ 成立

若 $a\times b<0$

不妨设 $a>0,b<0$且$|a|>|b|$

则 $|a|+|b|=a-b>a+b>-a-b>-a+b$ 成立

所以公式得证。

$$\max\left\{|x_i-x_j|+|y_i-y_j|\right\}$$
$$=\max\left\{\max\left\{x_i-y_i+x_j-y_j,x_i-y_i-x_j+y_j,-x_i+y_i+x_j-y_j,-x_i+y_i-x_j+y_j\right\}\right\}$$

定义集合

$$A=\{x_i-y_i+x_j-y_j\}$$
$$B=\{x_i-y_i-x_j+y_j\}$$
$$C=\{-x_i+y_i+x_j-y_j\}$$
$$D=\{-x_i+y_i-x_j+y_j\}$$
$$X=A\cup B\cup C\cup D$$

改变搜索最大值的顺序

$$\max\{\max\{x_i-y_i+x_j-y_j,x_i-y_i-x_j+y_j,-x_i+y_i+x_j-y_j,-x_i+y_i-x_j+y_j\}\}$$
$$=\max\{\max A,\max B,\max C,\max D\}$$
$$=\max X$$

命题成立

这样我们就完美的解决了二维情况下的问题。

- **第二步**

对于二元组这个方法是正确的。对于三维的情况，通过类比的方法，可想而知方法也是正确的，那么我们可以把这个方法推广到三维、四维以及五维中（要求 $2^d<<N$，d 为维数），这样这道题目我们就完美的解决了！！！

例题四：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。

分析：

- **第一步**

读完这道题目，通过异或的性质，会很自然的得到纯枚举的策略。直接枚举起始点 i 和终结点 j。

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

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

但是直接枚举起始点和终结点的时间复杂度是 $O(N^2)$，由于题目给出的范围是 100000 所以这个算法是不可以接受的，那么应该怎样解决呢？

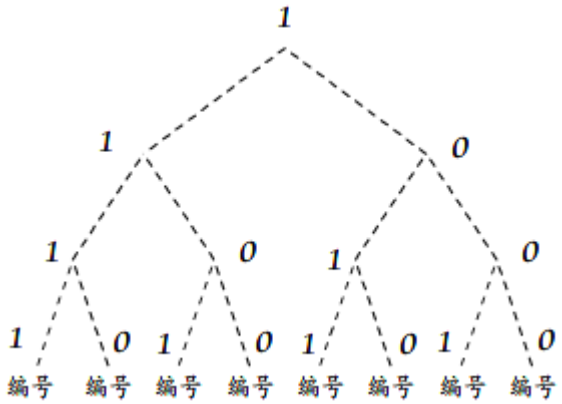
- **第二步**

应用二进制思想：

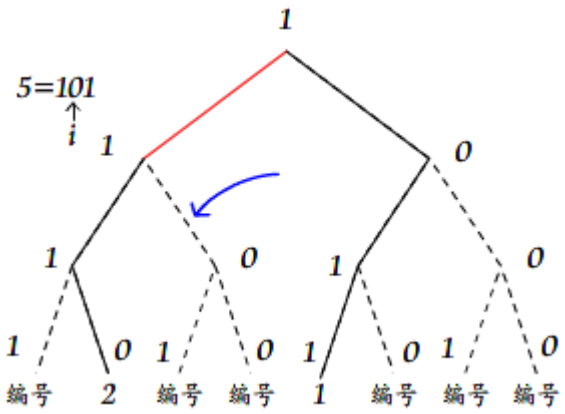
这道题目和以往的题目中最大的不同在于要求的是求得是异或值最大的子段而不是和或积，而且我们发现这些数的范围在 $0..2^{21}-1$ 的整数，也就是说把这些数转化成二进制的话，每个数最多只有 21 位，那么这个信息对我们有什么用呢？

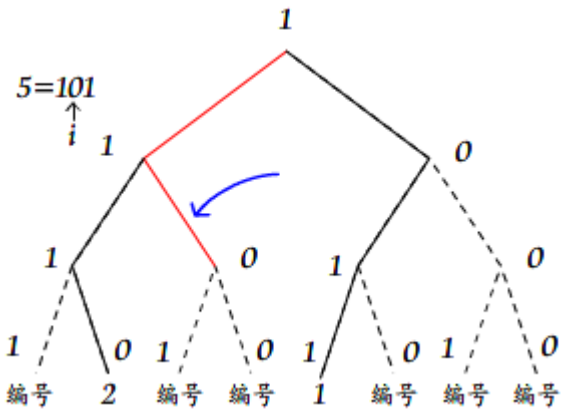
这样我们就可以用这一性质来建一棵高度为 22 的 01 二叉树，树根是 1，每个点的左儿子是 1，右儿子是 0.（由于篇幅限

制，仅用高度四层的模型代替）。



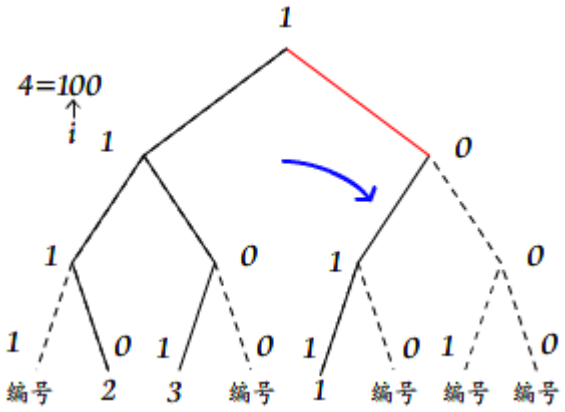
每次插入 $Sum_i=(x_1x_2...x_{20}x_{21})_2$时，从根开始往这棵树上添加一条长度为 21 的链，保证首尾相连，在第 i层的节点添加一条值为 x_i的边，叶子节点记录此数的编号，并且每次更新叶子节点的数值：

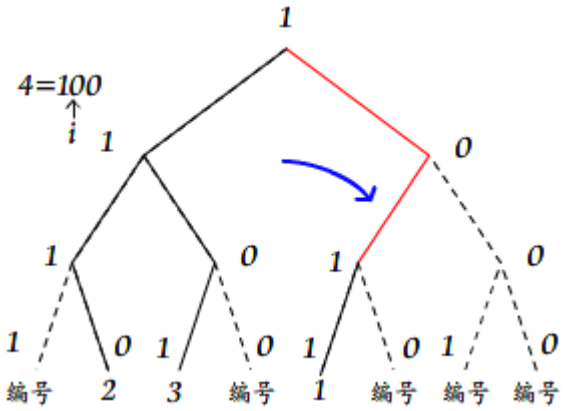




查询的时候怎么办呢？

查询的时候仅需根据当前插入的 Sum_i 进行贪心求解（如果能走不同的就走不同的）即可得出当前以 i 为终结点的最大值。





这样贪心是否正确呢？

证明：

$$Sum_i = (x_1 x_2 ... x_{n-1} x_n)_2$$

$$Sum_i = x_1 \times 2^{20} + x_2 \times 2^{19} + ... + x_{n-1} \times 2^1 + x_n \times 2^0$$

如果从高位往低位贪心的过程中，进行到第 i位时，如果有与此值不同的路可以走，则这样选取那条路之后，两个值（ Sum_i 与可选的另一路上的值）进行异或，则异或的值中必定含有 2^{x_i}，并且 $2^x > \sum_{i=0}^{x-1} 2^i$，所以这个贪心是正确的。

每次维护这棵 01 树，进行插入和查询，即可得到最后的答案，这样的时间复杂度为 O(NlogN).这样就可以完美的解决这道题目了！！！

总结

本文通过几个例子说明了二进制思想在信息学竞赛中的应用。在数据结构中，不仅巧妙地设计出了一种新的数据结构，而且既操作简单，应用方便，又时间空间复杂度不逊色于其他任何数据结构，更是把数据结构难于向多维拓展成为可能。树状数组在信息学竞赛中已经占有了一席地位。

在解题中，将二进制思想引入，运用十进制数与二进制数之间的关系，不仅可以用于状态压缩，还可以用与构建新的数学模型，不仅可以优化算法，还可以降低编程的复杂度。从而达到转十为二，事半功倍的效果！

所以说，二进制的方法不仅仅可以使算法得到优化，更是一种解题思想。

参考文献

1. 《算法艺术与信息学竞赛》，刘汝佳、黄亮著，清华大学出版社，2004 年 1 月第一版

2. Peking University Judge Online http://acm.pku.edu.cn/

3. ZheJiang University Judge Online http://acm.zju.edu.cn/

4. USA Computing Olympiad http://ace.delos.com/

附录

例题一原题：

Matrix

Description

Given an N*N matrix A, whose elements are either 0 or 1. A[i, j] means the number in the i-th row and j-th column. Initially we have A[i, j] = 0 (1 <= i, j <= N).

We can change the matrix in the following way. Given a rectangle whose upper-left corner is (x1, y1) and lower-right corner is (x2, y2), we change all the elements in the rectangle by using "not" operation (if it is a '0' then change it into '1' otherwise change it into '0'). To maintain the information of the matrix, you are asked to write a program to receive and execute two kinds of instructions.

1. C x1 y1 x2 y2 (1 <= x1 <= x2 <= n, 1 <= y1 <= y2 <= n) changes the matrix by using the rectangle whose upper-left corner is (x1, y1) and lower-right corner is (x2, y2).
2. Q x y (1 <= x, y <= n) querys A[x, y].

Input

The first line of the input is an integer X (X <= 10) representing the number of test cases. The following X blocks each represents a test case.

The first line of each block contains two numbers N and T (2 <= N <= 1000, 1 <= T <= 50000) representing the size of the matrix and the number of the instructions. The following T lines each represents an instruction having the format "Q x y" or "C x1 y1 x2 y2", which has been described above.

Output

For each querying output one line, which has an integer representing A[x, y].

There is a blank line between every two continuous test cases.

Sample Input

```
1
```

```
2 10
C 2 1 2 2
Q 2 2
C 2 1 2 1
Q 1 1
C 1 1 2 1
C 1 2 1 2
C 1 1 2 2
Q 1 1
C 1 1 2 1
Q 2 1
```

Sample Output

```
1
0
0
1
```

Source

POJ Monthly,Lou Tiancheng

例题二原题：

Sudoku

Description

In the game of Sudoku, you are given a large 9 × 9 grid divided into smaller 3 × 3 subgrids. For example,

.	2	7	3	8	.	.	1	.
.	1	.	.	.	6	7	3	5
.	.	.	.	.	.	.	2	9
3	.	5	6	9	2	.	8	.
.	.	.	.	.	.	.	.	.
.	6	.	1	7	4	5	.	3
6	4	.	.	.	.	.	.	.
9	5	1	8	.	.	.	7	.
.	8	.	.	6	5	3	4	.

Given some of the numbers in the grid, your goal is to determine the remaining numbers such that the numbers 1 through 9 appear exactly once in (1) each of nine 3 × 3 subgrids, (2) each of the nine rows, and (3) each of the nine columns.

Input

The input test file will contain multiple cases. Each test case consists of a single line containing 81 characters, which represent the 81 squares of the Sudoku grid, given one row at a time. Each character is either a digit (from 1 to 9) or a period (used to indicate an unfilled square). You may assume that each puzzle in the input will have exactly one solution. The end-of-file is denoted by a single line containing the word “end”.

Output

For each test case, print a line representing the completed Sudoku puzzle.

Sample Input

```
.2738..1..1...6735.......293.5692.8...........6.1
745.364.......9518...7..8..6534.
......52..8.4......3...9...5.1...6..2..7........3.
....6...1..........7.4.......3.
end
```

Sample Output

```
527389416819426735436751829375692184194538267268 1
745936432179589518436727829653 41
```

416837529982465371735129468571298643293746185864351297647913852359682714128574936

Source

Stanford Local 2006

例题三原题：

Requirements

Time Limit: 5 Seconds Memory Limit: 32768 KB

An undergraduate student, realizing that he needs to do research to improve his chances of being accepted to graduate school, decided that it is now time to do some independent research. Of course, he has decided to do research in the most important domain: the requirements he must fulfill to graduate from his undergraduate university. First, he discovered (to his surprise) that he has to fulfill 5 distinct requirements: the general institute requirement, the writing requirement, the science requirement, the foreign-language requirement, and the field-of-specialization requirement. Formally, a requirement is a fixed number of classes

that he has to take during his undergraduate years. Thus, for example, the foreign language requirement specifies that the student has to take 4 classes to fulfill this requirement: French I, French II, French III, and French IV. Having analyzed the immense multitude of the classes that need to be taken to fulfill the different requirements, our student became a little depressed about his undergraduate university: there are so many classes to take...

Dejected, the student began studying the requirements of other universities that he might have chosen after high school. He found that, in fact, other universities had exactly the same 5 requirements as his own university. The only difference was that different universities had different number of classes to be satisfied in each of the five requirement.

Still, it appeared that universities have pretty similar requirements (all of them require a lot of classes), so he hypothesized that no two universities are very dissimilar in their requirements. He defined the dissimilarity of two universities X and Y as |x1 - y1| + |x2 - y2| + |x3 - y3| + |x4 - y4| + |x5 - y5|, where an xi (yi) is the number of classes in the requirement i of university X (Y) multiplied by an appropriate factor that measures hardness of the corresponding requirement at the corresponding university.

Input

There are several test cases. The first line of each case contains an integer n (1 <= N <= 100000), the number of considered universities. The following N lines each describe the requirements of a university. A university X is described by the five non-negative real numbers x1 x2 x3 x4 x5. The input ends up with a case N = 0

Output

For each test case, on a single line, print the dissimilarity value of the two most dissimilar universities. Your answer should be rounded to exactly two decimal places.

Sample Input

```
3
2 5 6 2 1.5
1.2 3 2 5 4
7 5 3 2 5
0
```

Sample Output

```
12.80
```

Source: **MIT Programming Contest, 2005.02.26**

例题四原题：

Cow XOR

Adrian Vladu -- 2005

Farmer John is stuck with another problem while feeding his cows. All of his N ($1 \leq N \leq 100{,}000$) cows (numbered 1..N) are lined up in front of the barn, sorted by their rank in their social hierarchy. Cow #1 has the highest rank; cow #N has the least rank. Every cow had additionally been assigned a non-unique integer number in the range 0..(2^{21} - 1).

Help FJ choose which cows will be fed first by selecting a sequence of consecutive cows in the line such that the bitwise "xor" between their assigned numbers has the maximum value. If there are several such sequences, choose the sequence for which its last cow has the highest rank. If there still is a tie, choose the shortest sequence.

PROGRAM NAME: cowxor

INPUT FORMAT

- Line 1: A single integer N

- Lines 2..N+1: N integers ranging from 0 to 2^{21} - 1, representing the cows' assigned numbers. Line j describes cow of social hierarchy j-1.

SAMPLE INPUT (file cowxor.in)

```
5
1
0
5
4
2
```

INPUT DETAILS:

There are 5 cows. Cow #1 had been assigned with 1; cow #2 with 0; cow #3 with 5; cow #4 with 4; cow #5 with 2.

OUTPUT FORMAT

- Line 1: Three space-separated integers, respectively: the maximum requested value, the position where the sequence begins, the position where the sequence ends.

SAMPLE OUTPUT (file cowxor.out)

```
6 4 5
```

OUTPUT DETAILS:

4 xor 2 = 6 (001) xor (010) = (011)