原 线性基详解

2018年11月03日 11:07:28 Hypoc_ 阅读数 4065 更多

线性基是啥?
线性基是一个**数的集合**，并且每个序列都拥有**至少一个**线性基，取线性基中若干个数异或起来可以得到原序列中的任何一个数。

线性基三大性质

1. **原序列里面的任意一个数都可以由线性基里面的一些数异或得到**
2. **线性基里面的任意一些数异或起来都不能得到0**
3. **线性基里面的数的个数唯一，并且在保持性质一的前提下，数的个数是最少的**

线性基的构造

那么它是怎么构造的呢?
我们设有一个数组d，表示序列a的线性基，下标从0开始算。对于序列里面的每一个数，我们尝试将它插入到线性基里面去，具体如何插入这里给出代码好理解呀qwq，为了方便理解，我们设$x_{(2)}$为x的二进制数）：

```
for i=60 to 0
    if(x(2)的第i+1位为1)
    {
        if(d[i]为0)
        {
            d[i]=x;
            break;
        }
        else x=x^d[i];
    }
```

据此，我们可以得到一个关于d数组的性质：若$d[i]$不为0，则$d[i]_{(2)}$的第$i+1$位为1，并且$d[i]_{(2)}$的最高位就是第$i+1$位。
为了更好地进行线性基的讲解，我们要先知道**关于异或的一个小性质**（巨佬们可以跳过qwq）：
如果满足a ^ b ^ c=0，那么a ^ b=c，所以如果a ^ b=c，那么a ^ c=b
证明虽然简单但是这里就不给出了，不明白的读者手动模拟一下就明白了~~其实就是怕加上证明后博客变得过于臃肿~~

证明性质1

我们知道了线性基的构造方法后，其实就可以很容易想到如何证明性质1了，我们设原序列里面有一个数x，我们尝试用它来构造线性基，那么会有两种情况——1、不能成功插入线性基；2、成功插入线性基。

分类讨论一下

1、不能成功插入线性基**

什么时候不能插入进去呢?
显然就是它在尝试插入时异或若干个数之后变成了0。
那么就有如下式子：
x ^ $d[a]$ ^ $d[b]$ ^ $d[c]$ ^... $=0$
根据上面的那个小性质，则有：
$d[a]$ ^ $d[b]$ ^ $d[c]$ ^... $=x$
所以，如果x不能成功插入线性基，一定是因为当前线性基里面的一些数异或起来可以等于x。

2、可以成功插入线性基

我们假设x插入到了线性基的第i个位置，显然，它在插入前可能异或若干个数，那么就有：
x ^ $d[a]$ ^ $d[b]$ ^ $d[c]$ ^ ...=$d[i]$

$d[i]$ ^ $d[a]$ ^ $d[b]$ ^ $d[c]$ ^ ...$=x$
所以显然，x此时也可以由线性基里面的若干个数异或得到。

综上，性质1得证

再看性质2

各位大佬肯定认为这是一条显然的性质啊，但为了严谨一点，还是给出证明吧：
我们使用**反证法**
设$d[a]$ ^ $d[b]$ ^ $d[c]=0$（其中$d[c]$比$d[a]$和$d[b]$要更晚被插入线性基）
那么有$d[a]$ ^ $d[b]=d[c]$
$\because d[c]$可以由$d[a]$ ^ $d[b]$得到
$\therefore d[c]$不可能插入线性基

故假设不成立，所以线性基中不存在有任何数异或起来可以得到0。

最后看性质3

这个性质被BJWC拿来出过一道题，那题网上不知道为什么没有证明（都是草草的给出做法然后贴代码），然而如果你熟记这个性质3，那么很快就
题。（这题在文章末尾会给出）
那么我来尝试**证明性质3**（毕竟网上几乎找不到证明作为参考，这里笔者我只能乱推了）：

还是没什么卵用地分类讨论一下

1、假如序列里面的所有元素都可以插入到线性基里面

显然如果是这种情况的话，不管是用什么顺序将序列里的数插入到线性基里，线性基中的元素一定与原序列元素数量相同。所以性质3成立。

2、假如序列里面的一些元素不能插入到线性基里面

我们设x不能插入到线性基里面，那么一定满足形如$d[a]$ ^ $d[b]$ ^ $d[c]=x$的式子，那我们尝试将插入顺序改变，变成：$d[a]$、$d[b]$、x、$d[c]$。那
d[c]是不可能插入成功的，简单的证明：
$\because d[a]$ ^ $d[b]$ ^ $d[c]=x$
$\therefore d[a]$ ^ $d[b]$ ^ $x=d[c]$（根据上面那条并没有什么卵用的异或性质）
原来是x插入不进去，改变顺序后，$d[c]$插入不进去，也就是说，对于插入不进去的元素，改变插入顺序后，要么还是插入不进去，要么就是插入
另一个原来插入的进去的元素插入不进去了，所以，可以插入进去的元素数量一定是固定的。
显然，如果你去掉线性基里面的任意一个数，都会使得原序列里的一些（或一个）数无法通过用线性基里的元素异或得到，所以，每一个元素都是
句话说，这里面没有多余的元素，所以，这个线性基的元素个数在保持性质1的前提下，一定是最少的。
顺便贴上插入的代码（代码里的ll都是指$longlong$）：

```
void add(ll x)
{
    for(int i=50;i>=0;i--)
    {
        if(x&(1ll<<i))//注意，如果i大于31，前面的1的后面一定要加ll
        {
            if(d[i])x^=d[i];
            else
            {
                d[i]=x;
                break;//记得如果插入成功一定要退出
            }
        }
    }
}
```

可能有人会问，如果有三个数$a,b,c(a<b<c)$，先插入c导致a和b都不能插入，但是$a+b>c$，将a,b插入到线性基里面，那么得到的
大吗？
是会更大，但是这种情况不存在。~~（没想到吧）~~
假如先插入c会导致a,b不能插入，那么有c ^ $a>0,c$ ^ $b>0$，于是有a ^ $b>0$，所以a,b此时不 时插入。

如何求最大值

完整的说，是如何求在一个序列中，取若干个数，使得它们的异或和最大。
首先构造出这个序列的线性基，然后从线性基的最高位开始，假如当前的答案异或线性基的这个元素可以变得更大，那么就异或它，答案的初值为
代码如下：

```
ll ans()
{
    ll anss=0;
    for(int i=50;i>=0;i--)//记得从线性基的最高位开始
    if((anss^d[i])>anss)anss^=d[i];
    return anss;
 }
```


为啥求解是个贪心的过程？
前面说过,$d[i]_{(2)}$的第$i+1$位一定为1，联想一下这里，无非就是两种情况：

1. $ans_{(2)}$的第i+1位为0
   对于这种情况，ans异或了$d[i]$之后一定会变大，那么我们就选择异或它。
   可能有人会问，ans异或完$d[i]$后虽然i+1位变成了1，但是后面的1~i位可能会受到影响啊。
   管它呢~
   无论后面怎么变，就算后面的$1\sim i$位都能变成1，贡献也没有第$i+1$位变成1的贡献大呀。
   所以我们要优先使最高位尽可能大。
2. ans$_{(2)}$的第i+1位为1
   如果ans异或了$d[i]$，那么第$i+1$位就会变成0，后面怎么异或也补救不了第$i+1$位变成0的损失了，按照上面的思想，要优先使最高位尽可i
   时，ans不能异或$d[i]$。

如何求最小值

显然的，最小值一定是最小的$d[i]$。（如果让最小的d[i]去异或其它的$d[i]$一定会让它变得更大，所以它自己就是最小的）
代码就~~懒得贴~~不贴了。（别告诉我你不会求最小值！）

如何求第k小的值

完整的说，应该是——从一个序列中取任意个元素进行异或，求能异或出的所有数字中第k小的那个。
首先，要对这个序列的线性基处理一下，对于每一个$d[i]$，枚举$j=i\ to\ 1$，如果$d[i]_{(2)}$的第j位为1，那么$d[i]$异或$d[j-1]$。
那么处理完一个线性基之后，应该大致是长这个样子的（x表示0或1）：
1xxxx0xxx0x
  1xxx0x
    1x
求解过程：将k先转成二进制，假如k的第i位为1，ans就异或上线性基中第i个元素（注意不是直接异或d[i-1]）。
代码如下：

```
void work()//处理线性基
{
    for(int i=1;i<=60;i++)
    for(int j=1;j<=i;j++)
    if(d[i]&(1ll<<(j-1)))d[i]^=d[j-1];
}
ll k_th(ll k)
{
    if(k==1&&tot<n)return 0;//特判一下，假如k=1，并且原来的序列可以异或出0，就要返回0，tot表示线性基中的元素个数，n表示序列长度
    if(tot<n)k--;//类似上面，去掉0的情况，因为线性基中只能异或出不为0的解
    work();
    ll ans=0;
    for(int i=0;i<=60;i++)
    if(d[i]!=0)
    {
        if(k%2==1)ans^=d[i];
        k/=2;
    }
}
```

回想上面的线性基处理过程，可以发现，处理完之后，线性基中的元素，作用其实都是提供自己最高 1，那么只要使提供出来的1可以和$k_{(2)}$
1对应，那么求出来的ans一定就是第k小的。

补充：想想就能知道，其实处理完之后的线性基其实也还是原序列的一个线性基，因为依然拥有上面的三个性质，要知道，一个序列的线性基不唯
素数量唯一而已。

如何判断一个数是否能被当前线性基中的元素异或得到

把它尝试插入进线性基里面去，假如可以插入，说明不能异或得到，假如插不进去，则说明可以异或 ~~（原理然而上面已经讲了）~~

线性基删除操作

没想到吧，这东西还支持删除操作。

在线

具体的问题是这样的：

> 给一个序列，有三种操作，一是往序列中插入一个数，二是删除这个序列中的一个数，三要求你维护这个序列的线性

插入很好解决，插就完了。

重点是删除操作，如果要删除的数x在线性基外，那么直接删掉即可，问题是假如它在线性基内，把他删掉之后可能序列中其他的数可以填进来。
现在讨论一下x在线性基内的做法：
没有在线性基中的数，一定是因为线性基中的若干个数可以异或得到他，那么可以记录一下不在线性基中的数都是由线性基中的哪些数异或得到的
个线性基外的数对应一个集合S，这个集合内就是线性基中那些异或起来可以得到他的数。
假如线性基外的某一个数的S中包含x，那么就不需要删除x，把这个数删除即可。
原因是删除这个数和删除x是等价的，因为这个数在线性基中是可以代替x的，那么就当这个数代替了x，然后把x给删掉，这样子删除之后线性基
所以可以删除这个数。
假如x不被线性基外的任何一个数的S包含，那么因为每一个线性基中的数加入进来的时候，可能会异或若干个原来就在线性基中的数，那么对于
一个数，另外造一个集合P，记录线性基中这个数插入进来的时候异或过哪些数。然后找到线性基中最小的并且P包含x的数，让他异或线性基中其他
可，这样就能消除x在线性基中的影响。

离线

上面的问题并没有强制在线，所以也可以离线做。

离线的话其实更简单。

我们可以找到线性基中对于每一个数，可以代替他们的那些数，那么可以使线性基优先存删除时间晚的，那么就消除了上面的 把他删掉之后可能序列中其
来 这样的问题，其余操作一样。

模板题
更多好题（难度非递增）：
BJWC 2011 元素 | 题解
SCOI 2016 幸运数字 | 题解
TJOI 2008 彩灯 | 题解