

# [2019 Multi-University Training Contest 1]

[比赛网址](#)

## 训练结果

- rank:算了这个别说了
- 完成情况 : 3/6/13

## 题解

### B Operation

补题

#### 题意

给了一个序列，要求实现两种操作

1. 给定 \$l,r\$ 求 \$a[l..r]\$ 种选出其中的一些值的最大异或和
2. 在序列的后面加一个 \$x\$

#### 题解

开始想到线段树套线性基，发现时间和空间都爆了。后发现我们可以记录 \$a[1..dotsi]\$ 的线性基，添加时候则从高位到低位，尽量用当前的基去替换之前的基，这样能使所有的基离 \$r\$ 更近。查询的时候只用位置大于 \$l\$ 的基。

```
#include<iostream>
#include<cstdio>
#include<algorithm>
#include<cstring>
#define ll long long
using namespace std;
int read()
{
    int k=0,f=1;char c=getchar();
    for(;!isdigit(c);c=getchar()) if(c=='-') f=-1;
    for(;isdigit(c);c=getchar()) k=k*10+c-'0';return k*f;
}
const int N=1000055;
int T,n,m,a[N],pos[N][35],base[N][35],Base[35],Pos[35];
void build(int p,int x)
```

```
{  
    for(int i=29;i>=0;i--)  
    {  
        if(x&(1<<i))  
        {  
            if(!Base[i])  
            {  
                Base[i]=x,Pos[i]=p;  
                return;  
            }  
            else  
            {  
                if(p>Pos[i])  
                    swap(p,Pos[i]),swap(x,Base[i]);  
            }  
            x^=Base[i];  
        }  
    }  
}  
int main()  
{  
    for(T=read();T;T--)  
    {  
        memset(Base,0,sizeof(Base));  
        memset(Pos,0,sizeof(Pos));  
        n=read();m=read();  
        for(int i=1;i<=n;i++)  
            a[i]=read();  
        for(int i=1;i<=n;i++)  
        {  
            build(i,a[i]);  
            for(int j=0;j<=29;j++)  
                base[i][j]=Base[j],pos[i][j]=Pos[j];  
        }  
        int lans=0;  
        for(int i=1;i<=m;i++)  
        {  
            int l,r,opt;  
            opt=read();  
            if(!opt)  
            {  
                l=(read())^lans%n+1;  
                r=(read())^lans%n+1;  
                if(l>r) swap(l,r);  
                int ans=0;  
                for(int j=29;j>=0;j--)  
                {  
                    if(pos[r][j]>=l)  
                    {  
                        if(l>r) swap(l,r);  
                        l++;  
                    }  
                }  
            }  
        }  
    }  
}
```

```
        if((ans^base[r][j])>ans)
            ans=ans^base[r][j];
    }
}
printf("%d\n",lans=ans);
}
else
{
    l=read()^lans;
    n++;a[n]=l;
    build(n,a[n]);
    for(int j=0;j<=29;j++)
        base[n][j]=Base[j],pos[n][j]=Pos[j];
}
}
return 0;
}
```

## D Vacation

From:

<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:

[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:die\\_java:front\\_page\\_springtraining5&rev=1589552551](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:die_java:front_page_springtraining5&rev=1589552551)

Last update: 2020/05/15 22:22

