

Codeforces Round #704 (Div. 2)

[比赛链接](#)

E. Enormous XOR

题意

定义 $F(l,r)$ 表示 $l \oplus (l+1) \oplus \dots \oplus r$ 求 $\max_{L \leq l \leq r \leq R} F(l,r)$

题解

设 R 的长度为 n 从低位到高位依次编号 $0,1,2 \dots n-1$ 若 L 的第 $n-1$ 位为 0 ，取 $l=2^{n-1}-1, r=2^n-1$ 得 $F(l,r)=2^n$

若 L 的第 $n-1$ 位为 1 ，下面证明若 R 为偶数且 $R-L \geq 2$ 时答案为 $R+1$ 否则答案为 R

首先考虑 $R-L \geq 2$ 的情况，不难发现答案一定是 R

接下来考虑 R 是奇数的情况，考虑数学归纳法。

对固定的 $L \leq [L,L]$ 答案为 L $[L,L+1]$ 答案为 $L+1$ 同时 $L, L+1$ 其中一定有一个为奇数，满足数学归纳法边界条件。

假设 $[L,R]$ 答案为 R' 考虑 $[L,R+2]$ 的答案，对 $r \leq R$ 根据归纳法有 $F(l,r) \leq R$

对 $F(l,R+1)$ 假设 $F(l,R+1) > R+2$ 不妨假设 $F(l,R+1)$ 从高到低首个与 $R+2$ 不同的位为 k

此时有 $F(l,R+1)$ 第 k 位为 1 $R+2$ 第 k 位为 0 ，由于 $R+2$ 是奇数，所以 $k \geq 0$

不妨假设 $[l,R+1]$ 中最大的且第 k 位为 1 的数是 t

于是有 $[t+1,R+1]$ 第 k 位为 0 $[t-2^{k+1},t]$ 第 k 位为 1 $[t-2^{k+1}+1, t-2^k]$ 第 k 位为 $0 \dots$

由于 t 的末尾一定是 $111 \dots 1$ 于是 t 一定是奇数，于是 $[t+1,R+1]$ 长度为 $R-t+1$ 为奇数。

于是 $[l,R+1]$ 中有奇数个第 k 位为 1 和 $[l,R+1]$ 中有奇数个不能同时满足，即 $F(l,R+1)$ 第 k 位和第 $n-1$ 位不能同时为 1 ，矛盾。

对 $F(l,R+2)$ 有 $F(l,R+2) = F(l,R) \oplus 1 \leq R+1$

综上 R 为奇数时，答案不超过 R 同时 $F(R,R)$ 取到上界。

R 为偶数时，区间 $[L,R]$ 的答案一定不超过 $[L,R+1]$ 的答案。

由于 $R+1$ 是奇数，所以 $[L,R+1]$ 答案为 $R+1$ 于是 $R+1$ 是 $[L,R]$ 的答案上界，同时 $F(R-2,R) = R+1$ 取到上界。

```
const int MAXN=1e6+5;
int n;
char L[MAXN],R[MAXN];
void add(char *s){
    int pos=n-1;
    while(s[pos]=='1')s[pos]='0',pos--;
    s[pos]='1';
}
int main()
{
    n=read_int();
    scanf("%s%s",L,R);
    if(L[0]!=R[0]){
        _for(i,0,n)
            putchar('1');
    }
    else if(strcmp(L,R)==0||R[n-1]=='1')
        puts(R);
    else{
        add(L);
        if(strcmp(L,R)==0)
            puts(R);
        else{
            add(R);
            puts(R);
        }
    }
    return 0;
}
```

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

Permanent link: https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string;jxm2001:contest:cf_705_div_2&rev=1615086912 

Last update: 2021/03/07 11:15