

CCPC Wannafly Camp Day1

比赛链接

B. 萨博的方程式

题意

给定方程 $x_1 \oplus x_2 \oplus \dots \oplus x_n = k$

其中限定 $x_i \leq m_i$ 问方程的解的个数。

题解 1

不妨设 $m = \max(m_1, m_2, \dots, m_n)$ 且 m 的最高位是第 y 位。

假设 $\text{dp}(i, j)$ 表示前 i 个数有 j 个数第 y 位为 1 时 (x_1, x_2, \dots, x_i) 的取值方案数。

于是若 m_i 第 y 位为 1 ，分别考虑 x_i 第 y 位为 0 和为 1 的情况，有

$$\text{dp}(i, j) \text{ gets } 2^y \text{dp}(i-1, j) + (m_i \bmod 2^y) \text{dp}(i-1, j+1)$$

若 m_i 第 y 位为 0 ，有

$$\text{dp}(i, j) \text{ gets } (m_i \bmod 2^y) \text{dp}(i-1, j)$$

设 m_i 中有 cnt 个数第 y 位为 1 ，接下来我们考虑 $\text{dp}(n, i) (0 \leq i \leq \text{cnt})$ 对答案的贡献。

首先，如果 k 第 y 位为 0 ，则 i 的必须是偶数，如果 k 第 y 位为 1 ，则 i 的必须是奇数。

接下来对所有合法的 $i \in [0, \text{cnt}]$ 且 $x_1 \oplus x_2 \oplus \dots \oplus x_n$ 的第 y 位一定与 k 相同，接下来只需要考虑前 $y-1$ 位。

假如 $i = \text{cnt}$ 合法，则这等价于所有 x 的第 y 位已经确定，于是可以忽略 m_i 和 k 的第 y 位，问题转化为原题的子问题。

接下来考虑剩下所有合法的 $i \neq \text{cnt}$ 由于 $i < \text{cnt}$ 于是至少有一个 x_j 满足 x_j 第 y 位不等于 1 且 m_j 第 y 位等于 1 。

于是 x_j 的前 $y-1$ 位无限制。固定其他所有 x 对 $x_j \in [0, 2^{y-1}]$ 恰有一个 x_j 使得 $x_1 \oplus x_2 \oplus \dots \oplus x_n = k$

于是此时 i 对答案的贡献为 $\frac{\text{dp}(n, i)}{2^y}$ 总时间复杂度 $O(n^2 \log m)$

ps. $\text{dp}(i, j)$ 实际上可以换成 $f(i) = \sum_{j=0, 2 \dots i} \text{dp}(i, j)$, $g(i) = \sum_{j=1, 2 \dots i} \text{dp}(i, j)$ 维护，时间复杂度降为 $O(n \log m)$

```
const int MAXN=55,Mod=1e9+7;
int n,k;
int a[MAXN],dp[MAXN][MAXN];
int inv(int a){
    int ans=1,k=Mod-2;
    while(k){
        if(k&1)ans=1LL*ans*a%Mod;
        a=1LL*a*a%Mod;
        k>>=1;
    }
    return ans;
}
int cal(int pos){
    if(pos<0)return 1;
    dp[0][0]=1;
    _for(i,1,MAXN)dp[0][i]=0;
    int cnt=0;
    _for(i,0,n){
        if((a[i]>>pos&1)==1){
            cnt++;
            dp[cnt][0]=1LL*dp[cnt-1][0]*(1<<pos)%Mod;
            _rep(j,1,cnt)
            dp[cnt][j]=(1LL*dp[cnt-1][j]*(1<<pos)+1LL*dp[cnt-1][j-1]*(a[i]%(1<<pos)+1))%Mod;
        }
        else{
            _rep(j,0,cnt)
            dp[cnt][j]=1LL*dp[cnt][j]*(a[i]%(1<<pos)+1)%Mod;
        }
    }
    int ans=0;
    for(int i=k>>pos&1;i<cnt;i+=2)
        ans=(ans+dp[cnt][i])%Mod;
    ans=1LL*ans*inv(1<<pos)%Mod;
    if((cnt&1)==(k>>pos&1))
        ans=(ans+cal(pos-1))%Mod;
    return ans;
}
int main()
{
    while(~scanf("%d%d",&n,&k)){
        _for(i,0,n)a[i]=read_int();
        enter(cal(30));
    }
    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:ccpc_wannafly_winter_camp_day2&rev=1617201345

Last update: **2021/03/31 22:35**

