CCPC Wannafly Camp Day5

比赛链接

J. Xor on Figures

题意

给定 $2^k\times 2^k$ 的一个二维 01 串。定义 $B(x,y)$ 表示将二维 01 串循环左移 x 格循环上移 y 格的新二维 01 串。

定义两个二维串之间的异或为对应位置异或。问任意个 $B(x_i,y_i)$ 异或能得到的所有不同的二维 01 串个数。

题解

直接把 $B(x,y)$ 拉成一维，然后高斯消元计算所有 2^{2k} 个 $B(x,y)$ 的秩即可。时间复杂度 $\left(2^{4k}+\frac {2^{6k}}{w}\right)$

```
const int MAXN=35,Mod=1e9+7;
int a[MAXN][MAXN];
bitset<MAXN*MAXN> x[MAXN*MAXN];
char s[MAXN];
int quick_pow(int a,int k){
    int ans=1;
    while(k){
        if(k&1)ans=1LL*ans*a%Mod;
        a=1LL*a*a%Mod;
        k>>=1;
    }
    return ans;
}
int main()
{
    int k=read_int(),n=1<<k;
    _for(i,0,n){
        scanf("%s",s);
        _for(j,0,n)
        a[i][j]=s[j]-'0';
    }
    _for(di,0,n)_for(dj,0,n)
    _for(i,0,n)_for(j,0,n)
    x[di*n+dj][i*n+j]=a[(i+di)%n][(j+dj)%n];
    n*=n;
    int rk=0;
```

```
    _for(i,0,n){
        if(!x[i][i]){
            _for(j,i+1,n){
                if(x[j][i]){
                    swap(x[i],x[j]);
                    break;
                }
            }
        }
        if(x[i][i])
        rk++;
        _for(j,i+1,n){
            if(x[j][i])
            x[j]^=x[i];
        }
    }
    enter(quick_pow(2,rk));
    return 0;
}
```