I. War of Inazuma (Hard Version)

题意

要求给 n 位二进制数染上黑白两色，使得黑白两色的二进制数个数不相等。同时每个二进制数向其他只有一位与自身不同的二进制数连边。

要求与每个二进制数相邻的同色二进制数不超过 $m=\lceil\sqrt n\rceil$

题解

将二进制数写成一个 $\lceil\frac nm\rceil\times m$ 的矩阵，最后一行不足的位置补 1，然后将二进制数分为四类：

$A.$ 有奇数个 1，至少存在一行全为 1

$B.$ 有偶数个 1，不存在一行全为 1

$C.$ 有奇数个 1，不存在一行全为 1

$D.$ 有偶数个 1，至少存在一行全为 1

A,B 染白色C,D 染黑色。下面证明方案合法：

只考虑白色的合法性，黑色的合法性是对称的。

首先 $A$$B$ 内部不连边，因为内部 1 奇偶性相同所有至少有两个位置不同。

只有仅一行全 1 的 A 才能和 B 连边，否则至少有两个位置不同。

对于 A 仅一行全 1 的数 u为了和 B 中的数 v 连边，需要 v 其他行与 u 完全相同，然后 v 在 u 的全 1 行只有一个位置是 0。

由于一行只有 m 个数，于是这样的 v 最多只有 m 个。

对与 B 中的每个数 u为了和 A 中的数 v 连边，需要 v 其他行与 u 完全相同，然后仅某一行是全 1 行。

由于只有 $\lceil\frac nm\rceil$ 行，因此这样的 v 最多只有 $\lceil\frac nm\rceil\le m$ 个。

最后有 $A+C=B+D=2^{n-1},B+C=(2^m-1)^{\lceil\frac nm\rceil}$于是 $A+B,C+D$ 都是奇数。

由于 $4\mid A+B+C+D$于是 $A+B\neq C+D$

```
bool check(int n,int m,int v){
    bool flag_cnt=false;
    bool flag_one=false;
    for(int i=0;i<n;i+=m){
        bool flag_line=true;
        int r=min(n,i+m);
```

```
        _for(j,i,r){
            int c=v&1;
            flag_line&=c;
            flag_cnt^=c;
            v>>=1;
        }
        flag_one|=flag_line;
    }
    return flag_cnt^flag_one;
}
int main()
{
    int n=read_int(),m=ceil(sqrt(n));
    _for(i,0,1<<n)
    putchar('0'+check(n,m,i));
    return 0;
}
```