半数集问题：

如果使用递归会发现时间复杂度太大且难以避免大量重复运算

打代码之前，不妨手动模拟一下：
n=0,n=1 时，答案显然是 1
n=2, ans=2;　　n=3,ans=2
n=4,ans=4;　　n=5,ans=4
n=6,ans=6;　　n=7,ans=6

很容易发现，2n 与 2n+1(n 为非负整数)的答案是一样的 这就是第一个规律

然后以 n=8 为例，手动模拟一下

一共有 10 组解

8
1 8　2 8　3 8　4 8

1 2 8　1 3 8　1 4 8　2 4 8

1 2 4 8

打出的东西很像一棵搜索树。。。

当把 8 和 8 下面的左三棵子树放在一起（即 8 和下面三列），并将所有的 8 都改成 7，能发现，得到了 n=7 时的所有解；

再把最右端的子树（即剩下的部分）中的所有 8 删去，得到了 n=4 时的所有解

就这样，可以得到一个递推式：

```
     f(n)=f(n-1)        //7=8-1

       +f(n/2)          //4=8/2
```

再结合之前发现的规律

就能得到：

```
n%2==0 时
    f(n)=f(n-1)+f(n/2)
n%2==1 时
    f(n)=f(n-1)
```

双色汉诺塔问题：

通过手动模拟发现，其实双色与单色移动方法相同，因为在最小的移动方法中同色的两个盘子是不可能在一起的，所以仍可以采用普通汉诺塔的递归方法解决双色汉诺塔问题，即将一个问题分治为三个子问题，$T(n)=2T(n-1)+1$。