消除游戏

列表 `arr` 由在范围 `[1, n]` 中的所有整数组成，并按严格递增排序。请你对 `arr` 应用下述算法：

- 从左到右，删除第一个数字，然后每隔一个数字删除一个，直到到达列表末尾。
- 重复上面的步骤，但这次是从右到左。也就是，删除最右侧的数字，然后剩下的数字每隔一个删除一个。
- 不断重复这两步，从左到右和从右到左交替进行，直到只剩下一个数字。

给你整数 `n` ，返回 `arr` 最后剩下的数字。

示例 1：

输入： `n = 9`

输出： `6`

解释：

`arr = [1, 2, 3, 4, 5, 6, 7, 8, 9]`

`arr = [2, 4, 6, 8]`

`arr = [2, 6]`

`arr = [6]`

示例 2：

输入： `n = 1`

输出： `1`

```python
class Solution:
    def lastRemaining(self, n: int) -> int:
        #模拟删除法 在实际内存上并未实现物理删除 只通过删除间隔查找 start 首元素位置
        #起始为 1
        start = 1
        #起始间隔为 1
        tmp = 1
        #起始操作数为 0
        k = 0
        while n > 1:
            k += 1
```

```
            #k 为奇数，正向删除，偶数则为逆向删除
            if k % 2:
                start += tmp
            else:
                #当 n 为奇数时，逆序删的时候首位元素会被删掉，偶数则不变
                if n % 2:
                    start += tmp
            #每操作一次删除的跳跃间隔都将翻倍
            tmp *= 2
            #每操作一次删除 长度减半
            n //= 2
        return start
```