

## 圆圈中最后剩下的数字

0,1,,n-1 这  $n$  个数字排成一个圆圈，从数字 0 开始，每次从这个圆圈里删除第  $m$  个数字。求出这个圆圈里剩下的最后一个数字。

例如，0、1、2、3、4 这 5 个数字组成一个圆圈，从数字 0 开始每次删除第 3 个数字，则删除的前 4 个数字依次是 2、0、4、1，因此最后剩下的数字是 3。

示例 1:

输入:  $n = 5, m = 3$

输出: 3

示例 2:

输入:  $n = 10, m = 17$

输出: 2

约瑟夫环问题求解思路:

我们将上述问题建模为函数  $f(n,m)$ ，该函数的返回值为最终留下的元素的序号。

首先，长度为  $n$  的序列会先删除第  $m\%n$  个元素，然后剩下一个长度为  $n-1$  的序列。那么，我们可以递归地求解  $f(n-1,m)$ ，就可以知道对于剩下的  $n-1$  个元素，最终会留下第几个元素，我们设答案为  $x = f(n-1, m)$ 。

由于我们删除了第  $m\%n$  个元素，将序列的长度变为  $n-1$ 。当我们知道了  $f(n-1,m)$  对应的答案  $x$  之后，我们也就可以知道，长度为  $n$  的序列最后一个删除的元素，应当是从  $m\%n$  开始数的第  $x$  个元素。因此有  $f(n,m) = (m\%n + x) \% n = (m + x) \% n$ 。

我们递归计算  $f(n,m), f(n-1,m), f(n-2,m), \dots$  直到递归的终点  $f(1,m)$ 。当序列长度为 1 时，一定会留下唯一的那个元素，它的编号为 0。

```
class Solution {
public:
    //约瑟夫公式
    //f(n,m)=[f(n-1,m)+m]%n
    int fun(int n,int m)
    {
        if(n==1)
            return 0;
        int x=fun(n-1,m);
        return (m+x)%n;
    }
    int lastRemaining(int n, int m) {
        return fun(n,m);
    }
};
```