约瑟夫问题

简介
所有人围成一圈，顺时针报数，每次报到q的人将被杀掉，被杀掉的人将从房间内被移走。然后从被杀掉的下一个人重新报数，继续报q，再清除，直到剩余一人

q = 2

对于q为2的情况，有个简单的方法。假设有2^n 个人，从1开始数，那么最后活下来的那个一定是1.

那么可以借用这个规律，假设有9个人，从1开始数，数到2，杀死，接下来到3的时候，相当于8个人从3开始数，那么根据上面的规律，活下来的就是3了（8==2^3)

规律为：当q==2时，2^n+t个人活下来的是第2t+1个人。

e.g. q==2，n<=1e100，可以用这个思路做。用大数，找到第一个小于等于2^x 的数，直接得ans==2*(n-2^x)+1

q != 2

设：n+1个数，每次杀第q个的最终结果为A(n+1)； n个每次杀第q个为A(n)
显然A(n+1)=(A(n)+q)%(n+1)，A(1)==0

证明：

n+1个人的游戏中，第一个杀掉的是k−1，那么第二个人是k−1+k 。但是我们可以把杀掉一个人后的情况，看成n个人的游戏的开始，也就是说k−1+k看成n个人的游戏中的k−1。所以有A(n+1)=(A(n)+q)%(n+1)

```
int f(int n,int q){
    if(n==1){
        return 0;
    }
    return (f(n-1,q)+q)%n;
}//若第一个编号为1，ans++即可
```

icpc2018沈阳 K

题意：

给出n个人的环，约瑟夫环背景，求第m个死的位置。

解析：
n+1个人第m个死的位置相当于n 个人第m−1个死的位置转k个下标。即A(n,m)=(A(n−1,m−1)+k)%n

当m较小的时候直接推就行了。

对于m=1e18,k=1e6的数据，显然A(n,m)=(A(n−1,m−1)+k)%n 的这个%n在很多时候没有用处。

A+x*k*<*n*+x

x*(*k−*1)*<n−A

x<(*k*−1)*n*−*A*

处理一下就可以了。

板子总结

1、题目：n个人，1至m报数，问最后留下来的人的编号。

公式：f(n,m)=(f(n−1,m)+m)%n，f(0,m)=0;

代码：复杂度O(n)

```
typedef long long ll;
ll calc(int n, ll m) {
    ll p = 0;
    for (int i = 2; i <= n; i++) {
        p = (p + m) % i;
    }
    return p + 1;
}
```

2、题目：n个人，1至m报数，问第k个出局的人的编号。(k<1e6)

公式： f(n,k)=(f(n−1,k−1)+m−1)%n+1

f(n−k+1,1)=m%(n−k+1)

if (f==0) f=n−k+1

代码：复杂度O(k)

```
ll call(ll n, ll m, ll k) { // (k == n) equal(calc)
    ll p = m % (n - k + 1);
    if (p == 0) p = n - k + 1;
    for (ll i = 2; i <= k; i++) {
        p = (p + m - 1) % (n - k + i) + 1;
    }
    return p;
}
```

3、题目：n个人，1至m报数，问第k个出局的人的编号。 (m<1e6)

代码：复杂度O(m∗log(m))

```
ll call2(ll n, ll m, ll k) {
    if (m == 1) return k;
    else {
        ll a = n - k + 1, b = 1;
        ll c = m % a, x = 0;
        if (c == 0) c = a;
        while (b + x <= k) {
            a += x, b += x, c += m * x;
            c %= a;
            if (c == 0) c = a;
            x = (a - c) / (m - 1) + 1;
        }
        c += (k - b) * m;
        c %= n;
        if (c == 0) c = n;
```

```
        return c;
    }
}
```