

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重新报数，继续报 q ，再清除，直到剩下一人

$q = 2$

对于 $q=2$ 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. $q=2$, $n \leq 1e100$, 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 $ans = 2 * (n - 2^x) + 1$

$q \neq 2$

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。

$$A + xk < n + x$$

$$x(k-1) < n - A$$

$$x < (k-1)n - A$$

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 cal1(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 cal2(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;
    }
}

```

4、题目：n个人，1至m报数，问编号为k的人是第几个出局的。

代码：复杂度O(n)

```

11 n,k; //可做n<=4e7, 询问个数<=100, 下标范围[0,n-1]
11 dieInXturn(int n,int k,int x){ //n个人, 报数k, 下标为x的人第几个死亡
    11 tmp=0;
    while(n){
        x=(x+n)%n;
        if(k>n)x+=(k-x-1+n-1)/n*n;
        if((x+1)%k==0){
            tmp+=(x+1)/k;break;
        }else{
            if(k>n){
                tmp+=x/k;
                11 ttmp=x;
                x=x-(x/n+1)*(x/k)+(x+n)/n*n-k;
                n-=ttmp/k;
            }else{
                tmp+=n/k;
                x=x-x/k;
                x+=n-n/k*k;
                n-=n/k;
            }
        }
    }
    return tmp;
}

```