快速幂

主讲人：邓哲也

快速幂的引入

给出 a，b， c，求出 a^b mod c.

a， b， c <= 10^9

快速幂

枚举 b，连续乘 b 次 a，每乘一次对 c 取模一次。

时间复杂度关于 b 线性。

每次都乘的是 a，重复运算太多，考虑分治。

快速幂

比如计算出 a, a^2, a^4, a^8, a^{16}, $\cdots$

对 b 进行二进制转换，比如 $b = 11 = 8 + 2 + 1$

故 $a^b = a^{11} = a^8 * a^2 * a$

因此，只要预处理出 $\log_2 10^9$ 个 a 的幂次

计算的时候分解成 $\log_2 b$ 个乘积即可。

快速幂代码实现

```
int pow(int a, int b, int c){
    int ans = 1;
    while (b) {
        if (b & 1) ans = 1LL * ans * a % c;
        b >>= 1;
        a = 1LL * a * a % c;
    }
    return ans;
}
```

快速幂加强

给出 a，b，c，求出 $a^b \bmod c$.

a，b，c <= 10^{18}

快速幂加强

沿用之前的做法，但是注意这里的模数是 10^{18}。

如果两个小于 10^{18} 的数相乘，long long也存不下。

考虑到幂次可以转化为连乘，乘法也可以转化为连加。

同样的我们可以得到一个“快速”乘。

快速幂

比如计算出 a，2a，4a，8a，16a，…

对 b 进行二进制转换，比如 $b = 11 = 8 + 2 + 1$

故 $ba = 11a = 8a + 2a + a$

因此，只要预处理出 $\log_2 10^{18}$ 个 a 的乘积

计算的时候分解成 $\log_2 b$ 个加法即可。

因为每次两个小于 10^{18} 的数相加，long long是可以存的。

快速幂代码实现

```
long long mul(long long a, long long b, long long c){
    long long ans = 0;
    while (b) {
        if (b & 1) ans = (ans + a) % c;
        b >>= 1;
        a = (a + a) % c;
    }
    return ans;
}
```

NOIP 2013 Day1 T1 转圈游戏

n 个小伙伴（编号从 0 到 n-1）围坐一圈玩游戏。按照顺时针方向给 n 个位置编号，从0 到 n-1。最初，第 0 号小伙伴在第 0 号位置，第 1 号小伙伴在第 1 号位置，……，依此类推。
游戏规则如下：每一轮第 0 号位置上的小伙伴顺时针走到第 m 号位置，第 1 号位置小伙伴走到第 m+1 号位置，……，依此类推，第n – m号位置上的小伙伴走到第 0 号位置，第n-m+1 号位置上的小伙伴走到第 1 号位置，……，第 n-1 号位置上的小伙伴顺时针走到第m-1 号位置。
现在，一共进行了 10^k 轮，请问 x 号小伙伴最后走到了第几号位置。

$1 < n < 1,000,000$，$0 < m < n$，$1 <= x <= n$，$0 < k < 10^9$。

NOIP 2013 Day1 T1 转圈游戏

答案显然就是 $(x + m * 10^k) \% n$

用快速幂计算 $10^k \% n$ 即可。

下节课再见