

基本思路:

1. 预处理出 $a^{2^0}, a^{2^1}, a^{2^2}, \dots, a^{2^{\log b}}$. 这 b 个数
2. 将 a^b 用 $a^{2^0}, a^{2^1}, a^{2^2}, \dots, a^{2^{\log b}}$ 这 b 种数来组合, 即组合成 $a^b = a^{2^{x_1}} \times a^{2^{x_2}} \times \dots \times a^{2^{x_t}} = a^{2^{x_1} + 2^{x_2} + \dots + 2^{x_t}}$ 即用二进制表示
为什么 b 可用 $a^{2^0}, a^{2^1}, a^{2^2}, \dots, a^{2^{\log b}}$ 这 b 个数来表示? ∵
二进制可以表示所有数, 且用单一用二进制表示时, b 单一表示最大可表示为二进制形式的 $2^{\log b}$

注意:

- $b \& 1$ 就是判断 b 的二进制表示中第0位上的数是否为1, 若为1, $b \& 1 = \text{true}$, 反之 $b \& 1 = \text{false}$ [还不理解? 进传送门](#)
 $b \& 1$ 也可以用来判断奇数和偶数, $b \& 1 = \text{true}$ 时为奇数, 反之 $b \& 1 = \text{false}$ 时为偶数

```
1  #include<iostream>
2  using namespace std;
3  long long qmi(long long a,int b,int p)
4  {
5      long long res=1;
6      while(b)//对b进行二进制化,从低位到高位
7      {
8          //如果b的二进制表示的第0位为1,则乘上当前的a
9          if(b&1) res = res * a % p;
10         //b右移一位
11         b >>= 1;
12         //更新a,a依次为a^{2^0},a^{2^1},a^{2^2},...,a^{2^{\log b}}
13         a = a * a % p;
14     }
15     return res;
16 }
17 int main()
18 {
19     int n;
20     cin>>n;
21     while(n-->0)
22     {
23         cin.tie(0);
24         ios::sync_with_stdio(false);
25         int a,b,p;
26         long long res=1;
27         cin>>a>>b>>p;
```

```
28     res = qmi(a,b,p);
29     cout<<res<<endl;
30 }
31 return 0;
32 }
```