

太戈编程
etiger.vip

信奥算法

快速幂

快速幂取模

输入3个正整数 x, p, m , $x \leq 10^4$, $m \leq 10^4$, $p \leq 10^{15}$ 。
输出 $x^p \bmod m$ 的值, 即 x 的 p 次方除以 m 的余数。

输入样例:

3 2 4

输出样例:

1

说明:

$3 * 3 \bmod 4 = 9 \bmod 4 = 1$

输入样例:

5 3 11

输出样例:

4

说明:

$5 * 5 * 5 \bmod 11 = 125 \bmod 11 = 4$

算法分析

$x=3, p=11, m=7$

求 $3^{11} \bmod 7$

i
 $3^i \bmod 7$

1	2	4	8
3	2	4	2

二进制
拆分

求11个3的乘积

$(11)_{10} = (\textcolor{teal}{1}0\textcolor{blue}{1}\textcolor{red}{1})_2$

3^1 乘 3^2 乘 3^8

求11个3的乘积模7

$3^1 \bmod 7$ 乘 $3^2 \bmod 7$ 乘 $3^8 \bmod 7$

再模7

算法步骤

$x=3, p=11, m=7$

求 $3^{11} \bmod 7$

i
 $3^i \bmod 7$

1	2	4	8
3	2	4	2

$p=11$ 是奇数
 p 的二进制
末位是1

$p=p/2$ 更新
为5是奇数

$p=p/2$ 更新
为2是偶数

$p=p/2$ 更新
为1是奇数

ans初始化1

ans乘3模7

ans乘2模7

ans不变

ans乘2模7

ans更新为3

ans更新为6

ans更新为5

代码

```
4 long long x,p,m;  
5 cin>>x>>p>>m;  
6 long long ans=1;  
7 while(p){  
8     if(p%2)  
9         ans=  
10        p/=2;  
11        x=  
12    }  
13 cout<<ans<<endl;
```

x不断平方

class
自定义类型

纯public

```
3 class Student{
4     public:
5         int score;
6         Student(int x){
7             score=x;
8         }
9         int add(){
10             score++;
11         }
12 };
13 int main(){
14     Student s(99);
15     s.add();
16     cout<<s.score<<endl;
17     return 0;
18 }
```

构造函数
无返回值
做初始化

class
+
public
效果类似
struct


```
3 class Student{
4     private:
5         int score; ←
6     public:
7         Student(int x){
8             score=x;
9         }
10        int add(){
11            score++;
12        }
13    };
14    int main(){
15        Student s(99);
16        s.add();
17        cout<<s.score<<endl; ←
18        return 0;
19    }
```

private
变量

private
变量无法
直接访问

可以通过
public
函数访问

缺省默认
private

private
函数无法
直接访问

```
3 class Student{
4     int score;
5     Student(int x){
6         score=x;
7     }
8     int add(){
9         score++;
10    }
11 };
12 int main(){
13     Student s(99);
14     s.add();
15     cout<<s.score<<endl;
16     return 0;
17 }
```

this指针

指向当前实例

```
3 class Student{
4     private:
5         int score;
6     public:
7         Student(int score){
8             this->score=score;
9         }
10        int add(int score){
11            this->score+=score;
12        }
13        int print(){
14            cout<<this->score<<endl;
15        }
16    };
17    int main(){
18        Student s(99);
19        s.print();
20        s.add(100);
21        s.print();
22        return 0;
23    }
```

矩阵
matrix

$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix}$$

```
3 typedef long long ll;
4 const ll SIZE=10;
5 class Matrix{
6     public:
7         ll r,c;
8         ll data[SIZE][SIZE];
9         Matrix(int _r,int _c){
10             r=_r;
11             c=_c;
12             memset(data,0,sizeof(data));
13         }
14         void print(){
15             cout<<r<<endl;
16             cout<<c<<endl;
17             for(ll i=0;i<r;++i,cout<<endl)
18                 for(ll j=0;j<c;++j)
19                     cout<<data[i][j]<<" ";
20         }
21     };
```

```
22 int main(){
23     Matrix A(2,2);
24     A.print();
25     A.data[0][0]=1; A.data[0][1]=1;
26     A.data[1][0]=1; A.data[1][1]=0;
27     A.print();
28     return 0;
29 }
```

矩阵乘法

$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 8 \\ 14 \end{bmatrix}$$

$$0 \times 1 + 1 \times 2 = 2$$

$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 8 & 3 \\ 14 & 5 \end{bmatrix}$$

$$4 \times 0 + 5 \times 1 = 5$$

$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 8 & 3 \\ 14 & 5 \end{bmatrix}$$

```
Matrix operator*(const Matrix&m)const{
    Matrix ans(r,m.c);
    for(ll i=0;i<r;++i)
        for(ll j=0;j<m.c;++j)
            for(ll k=0;k<c;++k)
                ans.data[i][j]+=data[i][k]*m.data[k][j];
    return ans;
}
```

```
31 Matrix A=Matrix(2,2);
32 A.data[0][0]=1; A.data[0][1]=1;
33 A.data[1][0]=1; A.data[1][1]=0;
34 Matrix B=Matrix(2,1);
35 B.data[0][0]=2;
36 B.data[1][0]=3;
37 Matrix C=A*B;
38 C.print();
```

矩阵乘法

时间复杂度

$$O(N^3)$$

N表示矩阵行列数上限

单位矩阵

$$I = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}$$

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$v = \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix}$$

$$I \times v = ?$$

斐波那契数列
Fibonacci

$$f_n = f_{n-1} + f_{n-2}$$

$$\begin{bmatrix} f_n \\ f_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \times \begin{bmatrix} f_{n-1} \\ f_{n-2} \end{bmatrix}$$

$$\begin{bmatrix} f_n \\ f_{n-1} \end{bmatrix} = A^{n-1} \times \begin{bmatrix} f_1 \\ f_0 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

A必须是方阵
才可以求次幂

```
40 Matrix A(2,2);
41 A.data[0][0]=1; A.data[0][1]=1;
42 A.data[1][0]=1; A.data[1][1]=0;

43 ll n;
44 cin>>n;
45 if(n<=1) {
46     cout<<1<<endl;
47     return 0;
48 }

49 Matrix f10(2,1);
50 f10.data[0][0]=1;
51 f10.data[1][0]=1;
52 Matrix fnn_1=(A^(n-1)) * f10;
53 cout<<fnn_1.data[0][0]<<endl;
```

```
25 Matrix operator^(ll p) const{
26     Matrix ans(r,c);
27     Matrix x=*this;
28     for(ll i=0;i<r;++i)
29         ans.data[i][i]=1;
30     while(p){
31         if(p&1)
32             ans=ans*x;
33         x=x*x;
34         p>>=1;
35     }
36     return ans;
37 }
```



现场挑战 快快编程2560

递推式

$$a_n = a_{n-1} + a_{n-3}$$

矩阵乘法

$$\begin{bmatrix} a_n \\ a_{n-1} \\ a_{n-2} \end{bmatrix} = \begin{bmatrix} \square \\ \square \\ \square \end{bmatrix} \times \begin{bmatrix} a_{n-1} \\ a_{n-2} \\ a_{n-3} \end{bmatrix}$$

```
40 Matrix A(3,3);
41
42
43
44 ll n;
45 cin>>n;
46 if(n<=3) {
47     cout<<1<<endl;
48     return 0;
49 }
50 Matrix a321(3,1);
51
52
53
54 Matrix ann_1n_2=
55 cout<<<<endl;
```

太戈编程

653

149

2560

拓展题

40