《CSP-J 初级组算法中数学》

Day02-初等数论(中)

主讲人：小武老师

初等数论

质因数分解、算术基本定理、同余的概念、孙子定理等。

课程大纲

1 初等数论 (上)
奇数、偶数、质数、合数、约数、倍数、因数、最小公倍数、最大公约数、欧几里得算法等

2 初等数论 (中)
算术基本定理、同余关系、孙子定理等

3 初等数论 (下)
质数判定、质数筛、埃氏筛法、线性筛法等

4 函数 (上)
坐标、函数图像、一次函数、变量与函数等

5 函数 (下)
二次函数、指数函数、对数函数、根式与指数幂、幂运算等

6 数列基础
等差数列、等比数列、递推公式、通项公式等

7 矩阵基础
一维矩阵、二维矩阵、矩阵的运算、转置、杨辉三角等

8 数及其运算
数的进制、二进制、八进制、十六进制、编码（ASCII）等

9 计数原理与排列组合 (上)
加法原理、乘法原理、排列与组合、再看杨辉三角等

10 计数原理与排列组合 (下)
捆绑法、插空法、CSP真题训练等

算术基本定理

引理1： 任何大于1的整数n都可以分解成素因数的连乘积，即：

$$n = p_1 p_2 \cdots p_m, m \geq 1$$

定理1： （算术基本定理）任何大于1的整数n有且只有一种方法将其分解成素因数的连乘积，即：

$$n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_m^{\alpha_m}, m \geq 1$$

整数惟一分解定理

标准分解式

算术基本定理

定理1：（算术基本定理）任何大于1的整数n有且只有一种方法将其分解成素因数的连乘积，即：

$$n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_m^{\alpha_m}, m \geq 1$$

$30 = 2 \times 3 \times 5$

$88 = 2 \times 2 \times 2 \times 11 = 2^3 \times 11$

$1024 = 2^{10}$

$99099 = 3^2 \times 7 \times 11^2 \times 13$

算术基本定理

例1　求以下合数的标准分解式（质因数分解）

$$18 = 2 \times 3^2$$

$$100 = 2 \times 5^2$$

$$105 = 3 \times 5 \times 7$$

$$1200 = 2^4 \times 3 \times 5^2$$

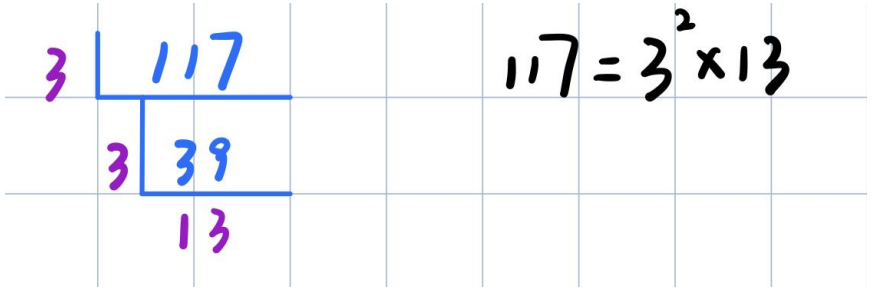
Q1: 有什么规律？能否描述计算的步骤？

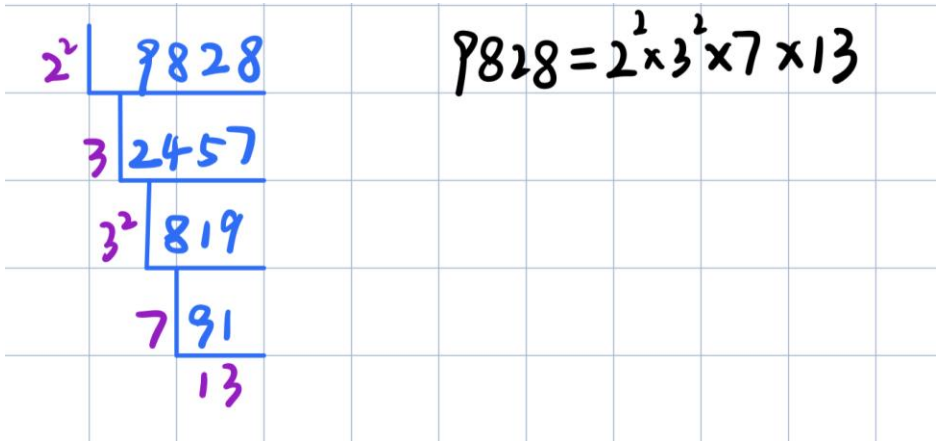
Q2: 如果编程来实现呢？

算术基本定理

定理1： （算术基本定理）任何大于1的整数n有且只有一种方法将其分解成素因数的连乘积，即：

$$n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_m^{\alpha_m}, m \geq 1$$





质因数分解

算术基本定理，又称为正整数的唯一分解定理，即：每个大于1的自然数，要么本身就是质数，要么可以写为2个或以上的质数的积，而且这些质因子按大小排列之后，写法仅有一种方式。

$$6936 = 2^3 \times 3 \times 17^2 \qquad 1200 = 2^4 \times 3 \times 5^2$$

算术基本定理的内容由两部分构成：

- 分解的存在性：
- 分解的唯一性，即若不考虑排列的顺序，正整数分解为素数乘积的方式是唯一的

把一个合数分解成若干个质因数的乘积的形式，即求质因数的过程叫做分解质因数。

P0067 质因数分解

把一个合数分解成若干个质因数的乘积的形式，即求质因数的过程叫做分解质因数。

$$1200 = 2^4 \times 3 \times 5^2$$

```
#include<bits/stdc++.h>
using namespace std;
void dfs(int n, int p){
    if(n==1) return;
    if(n%p == 0){
        cout << p <<" ";
        dfs(n/p,p);
    }else{
        dfs(n,p+1);
    }
}
```

```
int main(){
    int n;
    cin >> n;
    dfs(n,2);
    return 0;
}
```

同余

在日常生活中，我们所需要注意的常常不是某些整数，而是些整数用某一固定的正整数去除所得的余数。

同余的概念

问题 1： 比如从北京开往深圳的火车，20点55分开，全程的时间是14小时15分，那么几点到深圳呢？

$$35h10m = 24 \times 1 + 11h10m$$

问题 2：如果1978年的元旦是星期日，请问1979年的元旦是星期几？

$$365 = 7 \times 52 + 1$$

由于同是几点钟或同为星期几，常常在生活中有同样的意义，这样就在数学中产生了“同余”的概念。

同余关系

定义1： 如果a和b都是整数而m是一个固定的正整数，则当m|(a-b)(即m能够整除a-b)时，我们就说a,b对模m同余，记作：

$$a \equiv b(\mathrm{mod} m)$$

当m不能整数a-b时，则我们就说a,b对模m不同余，记作：

$$a \not\equiv b(\mathrm{mod} m)$$

$29 \equiv 2(\mathrm{mod}9)$

$93 \equiv -7(\mathrm{mod}50)$

$161 \not\equiv 0(\mathrm{mod}8)$

$257 \not\equiv 16(\mathrm{mod}32)$

同余关系

引理1：　如果a,b,c都是整数而m是一个正整数，则当

$$a \equiv b(\mathrm{mod}m)$$
$$b \equiv c(\mathrm{mod}m)$$

都成立时，我们有　$a \equiv c(\mathrm{mod}m)$

传递性

$$22 \equiv 13(\mathrm{mod}3)$$
$$13 \equiv 10(\mathrm{mod}3)$$

$$22 \equiv 10(\mathrm{mod}3)$$

两个整数a、b，若它们除以整数m所得的余数相等，则称a与b对于模m同余或a同余于b模m。

记作：a≡b (mod m)，

读作：a同余于b模m，或读作a与b对模m同余，例如 26≡2(mod 12)。

同余关系

同余式相加：若a≡b(mod m)，c≡d(mod m)，则a+c≡b+d(mod m)；

同余式相乘：若a≡b(mod m)，c≡d(mod m)，则ac≡bd(mod m)。

线性运算：如果a ≡ b (mod m)，c ≡ d (mod m)，那么

(1)a ± c ≡ b ± d (mod m)

(2)a * c ≡ b * d (mod m)

幂运算：如果a ≡ b (mod m)，那么a^n ≡ b^n (mod m)

$$22 \equiv 13 (\mathrm{mod} 3)$$
$$11 \equiv 8 (\mathrm{mod} 3)$$

$$33 \equiv 21 (\mathrm{mod} 3)$$
$$11 \equiv 5 (\mathrm{mod} 3)$$
$$242 \equiv 104 (\mathrm{mod} 3)$$

同余关系

求(a+b+c+...d)%m

将a,b,c..,d分解成z1*m+k1 , z2*m+k2 , z3*m+k3.....z4*m+k4则

原式=(z1*m+k1 , z2*m+k2 , z3*m+k3.....z4*m+k4)%m

=(k1+k2+k3+...k4)%m

=(a%m+b%m+c%m+....d%m)

乘法和乘方类似的。

推论：对于加法、乘法、乘方运算，算好后取余和边算边取余是等价的

同余关系

假设今天是星期日，那么a^b天之后是星期几？

```
int a, b;
cin >> a >> b;
int n = 1;
for(int i = 1; i <= b; i++)
    n = n*a;
```

```
int a, b;
cin >> a >> b;
int n = 1;
for(int i = 1; i <= b; i++)
    n = n*a%7;
```

Q3: 这样写有什么问题？

推论：对于加法、乘法、乘方运算，算好后取余和边算边取余是等价的

同余关系

a^b的末三位数字是多少?

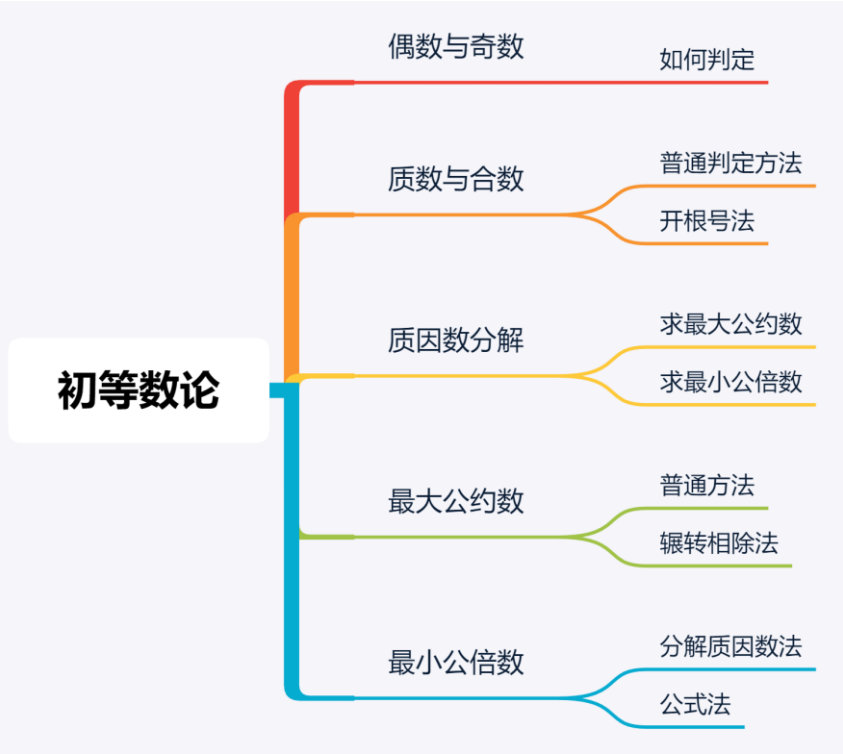
1011^3 最后末三位数字是多少?

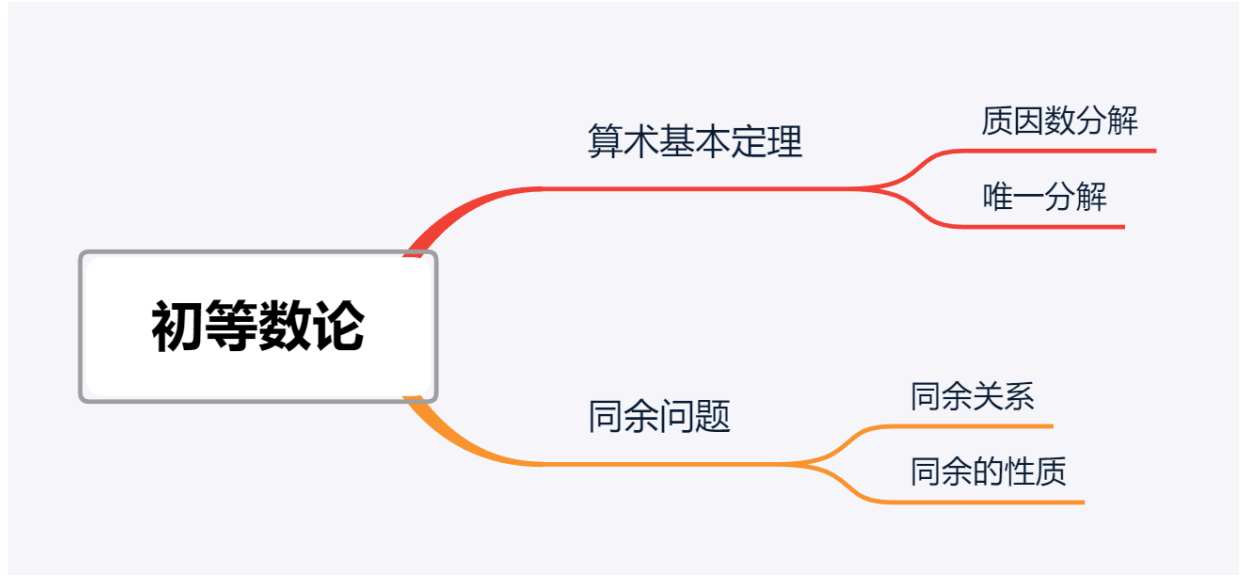
$$11 \times 11 \times 11 = 1331$$

```
int a, b;
cin >> a >> b;
int n = 1;
for(int i = 1; i <= b; i++)
    n = n*a%1000;
```

推论：对于加法、乘法、乘方运算，算好后取余和边算边取余是等价的

总结


初等数论
偶数与奇数
如何判定
质数与合数
普通判定方法
开根号法
质因数分解
求最大公约数
求最小公倍数
最大公约数
普通方法
辗转相除法
最小公倍数
分解质因数法
公式法


初等数论
算术基本定理
质因数分解
唯一分解
同余问题
同余关系
同余的性质

课后习题与实验

Talk is cheap, show me the code !

下节课见啦！