NCKU Programming Contest Training Course
Math
2018/03/14

Lin Yun Wen

Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan

made by kk & rabbit125
made by ai281918 & yun wen

Outline

Prime Numbers

Big Number

GCD, Extended Euclid's Algorithm

made by kk & rabbit125
made by ai281918 & yun wen

Prime Number

- Sieve of Eratosthenes (埃拉托斯特尼篩法)
  - 由小到大選擇質數，並刪除其倍數
- 6n±1 Method
  - 拿 2 和 3 這兩個質數先篩過一遍，剩下的數字則用除法驗證是不是質數。

made by kk & rabbit125
made by ai281918 & yun wen

Prime Number

- We use **sieve** to create a prime array
  - Chose the smallest number at each iteration and delete the multiple of this number

Chose 11

2 ~~3~~ ~~4~~ 5 ~~6~~ 7 ~~8~~ ~~9~~ ~~10~~ 11 ~~12~~ 13 ~~14~~ ~~15~~ ~~16~~ 17 ~~18~~ 19 ~~20~~ ~~21~~ ~~22~~…

made by kk & rabbit125
made by ai281918 & yun wen

Prime Number

- Sieve of Eratosthenes (埃拉托斯特尼篩法)
  - 由小到大選擇質數，並刪除其倍數

```cpp
#include <cmath>
#include <cstring>
#define MAX 10000000
bool is_prime[MAX];
void eratosthenes()
{
    memset(is_prime, 1, sizeof(is_prime));
    is_prime[0] = false;
    is_prime[1] = false;

    for (int i = 2; i <= sqrt(MAX); ++i)
        if (is_prime[i])
            for(int j = i+i; j < MAX; j += i)
                is_prime[j] = false;
}
```


made by kk & rabbit125
made by ai281918 & yun wen

Prime Number

- 6n±1 Method
- – 2 和 3 的最小公倍數是 6 ，把所有數字分為 6n 、 6n+1 、 6n+2 、
  - 6n+3 、 6n+4 、 6n+5 六種，可以看出 6n 、 6n+2 、 6n+3 、 6n+4 會是 2 或 3 的倍數，不屬於質數。因此，只要驗證 6n+1 和 6n+5 (= 6n-1)是不是質數就可以了。

Prime Number

- 6n±1 Method

```
#include <vector>
#define MAX 10000000
vector<int> prime;
bool is_prime(int n) {
    for (int i = 0; prime[i]*prime[i] <= n; ++i)
        if (n % prime[i] == 0)
            return false;
    return true;
}
void make_prime() {
    prime.push_back(2);
    prime.push_back(3);
    for (int i = 5, gap = 2; i < MAX; i+=gap, gap = 6 - gap)
        if (is_prime(i))
            prime.push_back(i);
}
```

made by kk & rabbit125
made by ai281918 & yun wen

Prime Number

- 方法二比方法一慢，但較省空間
- But just remember that the code in previous page is fast enough to solve almost every prime problems

- 其他方法:
  - 演算法 筆記 - Prime

Practice - 1

UVa 10392 - Factoring Large Numbers

made by kk & rabbit125
made by ai281918 & yun wen

Outline

Prime Numbers

Big Number

GCD, Extended Euclid's Algorithm

made by kk & rabbit125
made by ai281918 & yun wen

Big Number

- Array
- 習慣上將低位數放在index比較小的位置
  - Ex:68046897523 1245

0	1	2	3	4	5	6	7	8	9	…								
5	4	2	1	3	2	5	7	9	8	6	4	0	8	6	_	_	_	- - - -

- 右方補0

made by kk & rabbit125
made by ai281918 & yun wen

Big Number

- 加法：位數各自相加後，由低至高位依序進位
- 減法：位數各自相減後，由低至高位依序借位
- 乘法：直式乘法
- 除法：長除法

Big Number

- 加法：

```
void add(int a[100], int b[100], int c[100]) {
    for (int i = 0; i < 100; ++i)
        c[i] = a[i] + b[i];

    for (int i = 0; i < 100-1; ++i) {
        c[i+1] += c[i] / 10;
        c[i] %= 10;
    }
}
```

made by kk & rabbit125
made by ai281918 & yun wen

Practice - 2

UVa 10106 - Product

Problem Description

The problem is to multiply two integers X,Y .(0 ≤ X,Y < 10250)

Input

The input will consist of a set of pairs of lines. Each line in pair contains one multiplyer.

Output

For each input pair of lines the output line should consist one integer the product.

Outline

Prime Numbers

Big Number

GCD, Extended Euclid's Algorithm

made by kk & rabbit125
made by ai281918 & yun wen

Greatest Common Divisor

- 輾轉相除法 (Euclidean Algorithm)

```
int gcd(int a, int b) {
    if (a == 0)
        return b;
    return gcd(b % a, a);
}
```

- gcd(462, 1071)
  - gcd(147, 462)
    - gcd(21, 147)
      - gcd(0, 7)

```
int gcd(int a, int b) {
    if (a == 0)

    r
}
```

- 從1071中不斷減去462直到小於462（可以減2次，即商q0 = 2），餘數是147：
  - 1071 = 2 x 462 + 147.
- 然後從462中不斷減去147直到小於147（可以減3次，即q1 = 3），餘數是21：
  - 462 = 3 x 147 + 21.
- 再從147中不斷減去21直到小於21（可以減7次，即q2 = 7），沒有餘數：
  - 147 = 7 x 21 + 0.
- 此時，餘數是0，所以1071和462的最大公因數是21，

Practice - 3

UVa 408 – Uniform Generator

made by kk & rabbit125
made by ai281918 & yun wen

Extended Euclidean Algorithm

- 找到aX + bY = gcd(a,b)的整數解 X,Y
- Ex (from wiki)
  - 47x + 30y = 1

Extended Euclidean Algorithm

- 47 = 30 * 1 + 17
- 30 = 17 * 1 + 13
- 17 = 13 * 1 + 4
- 13 = 4 * 3 + 1
- 4 = 1 * 4 + 0

gcd(30, 47)
gcd(17, 30)
gcd(13, 17)
gcd(4, 13)
gcd(1, 4)
gcd(0, 1)

Extended Euclidean Algorithm

- $47 = 30 * 1 + 17$
- $30 = 17 * 1 + 13$
- $17 = 13 * 1 + 4$
- $13 = 4 * 3 + 1$
- $4 = 1 * 4 + 0$

- $17 = 47 * 1 + 30 * (-1)$
- $13 = 30 * 1 + 17 * (-1)$
- $4 = 17 * 1 + 13 * (-1)$
- $1 = 13 * 1 + 4 * (-3)$

$$47x + 30y = 1$$

Extended Euclidean Algorithm

- $1 = 13 * 1 + 4 * (-3)$
- $1 = 13 * 1 + [17 * 1 + 13 * (-1)] * (-3)$
- $1 = 17 * (-3) + 13 * 4$
- $1 = 17 * (-3) + [30 * 1 + 17 * (-1)] * 4$
- $1 = 30 * 4 + 17 * (-7)$
- $1 = 30 * 4 + [47 * 1 + 30 * (-1)] * (-7)$
- $1 = 47 * (-7) + 30 * 11$

made by kk & rabbit125
made by ai281918 & yun wen

Extended Euclidean Algorithm

- $gcd(a,b) = gcd(b,a\%b)$
- $aX + bY = gcd(a,b) = gcd(b,a\%b) = bX' + (a\%b)Y'$
- $aX + bY = bX' + [a - (a/b)b]Y' = aY' + b(X'-(a/b)Y')$
  - $X = Y'$
  - $Y = X' - (a/b)Y'$

Extended Euclidean Algorithm

```
int exGCD(int a, int b, int &X, int &Y) {
    if (b == 0) {
        X = 1;
        Y = 0;
        return a;
    } else {
        int gcd = exGCD(b, a % b, X, Y);
        int tmp = X;
        X = Y;
        Y = tmp - (a/b)*Y;
        return gcd;
    }
}
```

made by kk & rabbit125
made by ai281918 & yun wen

Practice - 4

UVa 10104 - Euclid Problem

made by kk & rabbit125
made by ai281918 & yun wen

Extended Euclidean Algorithm

- $\frac{m!}{n!}$ % P (P是一個很大的質數)

Extended Euclidean Algorithm

- $aX + bY = \gcd(a, b)$
  - $a = n!$
  - $b = p$
  - $\gcd(a,b) = 1$

方程式 ax+by=1 有整數解
iff 整數 a 和 b 互質

Extended Euclidean Algorithm

- n!X + pY = 1 (use Extended Euclidean Algorithm get (X,Y))
- n!X + pY = 1 → mod p
- (n!X) % p = 1 --- (1)
- $\frac{m!}{n!} \% \, p = ans$ --- (2)
- (1) * (2)
  $\rightarrow \left(\frac{m!}{n!} \times n!\,X \right) \% \, p = ans \rightarrow (\, m! \times X \,) \% \, p = ans$

made by kk & rabbit125
made by ai281918 & yun wen

Practice - 5

Facebook Hacker Cup 2017 Round1

Beach Umbrellas

made by kk & rabbit125
made by ai281918 & yun wen

Epsilon ε

- Float :
  - 數值範圍 :-3.4e-38 ~ 3.4e38
  - 十位數精確度位數 :6~7

- Double :
  - 數值範圍 :-1.7e308~1.7e308
  - 十位數精確度位數 :14~15

Epsilon ε

- Example

```
#include <cstdio>
#include <cmath>

int main() {
    double a = asin(sqrt(2.0) / 2) * 4.0;
    double b = acos(-1.0);

    printf("a = %.20lf\n", a);
    printf("b = %.20lf\n", b);
    printf("a-b = %.20lf\n", a - b);
    printf("a == b? %s\n", a == b ? "True" : "False");
}
```

made by kk & rabbit125
made by ai281918 & yun wen

Epsilon ε

- Result

```
linyunwen@Lin-Yun-Wens-MacBook-Air ~/D/L/c/ACM> ./sample_epsilon
a = 3.14159265358979356009
b = 3.14159265358979311600
a-b = 0.00000000000000044409
a == b? False
```

Epsilon ε

- 引入 eps 判斷浮點數是否相等
  - eps = 1e-8

整數	浮點數
a == b	|a – b| < eps
a != b	|a – b| > eps
a < b	a – b < -eps
a > b	a – b > eps

made by kk & rabbit125
made by ai281918 & yun wen

Practice - 6

UVa 906 – Rational Neighbor

made by kk & rabbit125
made by ai281918 & yun wen