NCKU Programming Contest Training Course
Math
2016/03/09

葉冠廷
xns77477@gmail.com

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

Greatest Common Divisor

Greatest Common Divisor

- 輾轉相除法 (Euclid's Algorithm)

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

Greatest Common Divisor

- 輾轉相除法 (Euclid's Algorithm)

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

- a x b = GCD(a, b) x LCM(a, b)
- GCB(a, b) = GCD(b, a % b)

Practice

UVA 408
Uniform Generator

Extended Euclid's Algorithm

- $aX + bY = GCD(a, b)$, find a particular solution of (X, Y).
- Ex: $12X + 9Y = 3$, $(X, Y) = (1, -1)$

Extended Euclid's Algorithm

- $aX + bY = GCD(a, b)$, find a particular solution of (X, Y).
- Ex: $12X + 9Y = 3$, $(X, Y) = (1, -1)$

- Theorem: $GCD(a, b) = GCD(b, a \% b)$

- $aX + bY = GCD(a, b) = GCD(b, a\%b) = bX' + (a\%b)Y'$

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

  $X = Y'$
  $Y = X' - (a/b)Y'$

Extended Euclid's 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;
    }
}
```

Extended Euclid's Algorithm

找出等式 AX + BY = N 的所有解:

1. 確認 N 是否能整除 GCD(A, B)
   - 若不能 => 無解
2. 等式兩邊同除以GCD(A, B)，變成
   aX + bY = n，其中 GCD(a, b) = 1
3. 使用 exGCD 找出特定解(x, y)
4. 則可以得到 X = n * x + b * t
   Y = n * y – a * t　其中 t 為整數

ex: 12X + 9Y = 9

4X + 3Y = 3

exGCD(4, 3) => (x, y) = (1, -1)

(X, Y) = (3 + 3t, -3 – 4t)

Practice

UVA 10104
Euclid Problem

Prime Number

Prime Number

- 用篩選方式建質數表
  - 由小到大選擇質數，並刪除其倍數

Prime Number

- 用篩選方式建質數表
  - 由小到大選擇質數，並刪除其倍數

```
#include <cstring>
#include <cmath>
#define MAX 1000000
bool isprime[MAX];
void Sieve()
{
    memset(isprime, true, sizeof(isprime));
    isprime[0] = isprime[1] = false;
    for (int i = 2; i <= sqrt(MAX); ++i)
        if (isprime[i])
            for (int j = i + i; j < MAX; j += i)
                isprime[j] = false;
}
```

Prime Number

- 質數從 5 開始後，只會出現在+2，+4，+2，+4…的數字上

Prime Number

- 質數從 5 開始後，只會出現在+2，+4，+2，+4…的數字上

- 5、7、11、13、17、19、21、23、27、29……

Prime Number


```
#define MAX 1000000
vector<int> prime;
bool isPrime(int n)
{
    for (int i = 0; prime[i] * prime[i] <= n; ++i)
        if (n % prime[i] == 0)
            return false;
    return true;
}
void MakePrime()
{
    prime.push_back(2);
    prime.push_back(3);
    for (int i=5, gap=2; i < MAX; i+=gap, gap=6-gap)
        if (isPrime(i)) prime.push_back(i);
}
```

Prime Number

- 方法二比方法一慢，但較省空間
- 其他方法:
  演算法筆記 - Prime
- 但是前頁的方法已經可以解幾乎所有的題目

Practice

UVA 10392
Factoring Large Numbers

Epsilon

Epsilon

Float:

- 數值範圍: -3.4e-38～3.4e38
- 十進位精確度位數: 6~7

Double:

- 數值範圍: -1.7e-308～1.7e308
- 十進位精確度位數: 14~15

Epsilon

Example:

```cpp
#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");
}
```

Epsilon

Result:

```
a = 3.1415926535897936000 0
b = 3.1415926535897931000 0
a - b = 0.0000000000000044409
a == b? False
```

Epsilon

引入eps，來判斷浮點數是否相等

eps = 1e-8

Epsilon

引入eps，來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

Practice

UVA 906
Rational Neighbor

Big Number

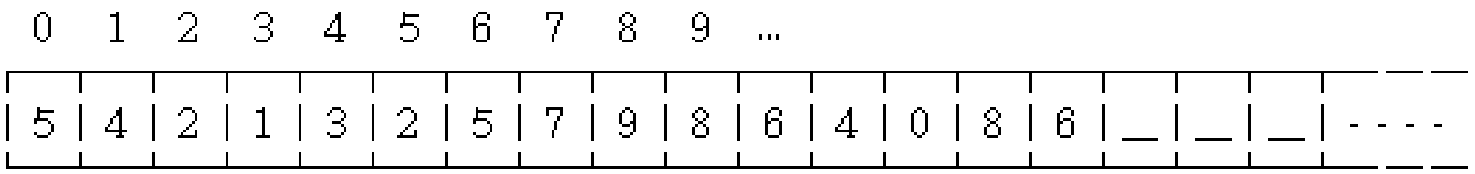

acm International Collegiate Programming Contest
IBM event sponsor

Big Number

- Array
- 習慣將低位數放在 index 比較小的位置

Big Number

- Array
- 習慣將低位數放在 index 比較小的位置
- ex: 680468975231245



- 右方補0

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

Ex: 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;
    }
}
```

Practice

UVA 10106
Product

Problem List

UVA

113, 11538, 10035, 338, 375, 10494, 746, 12404, 906, 10867, 10311, 11408, 10407, 10606, 10090