Thuật toán Euclid

Ngày 3 tháng 11 năm 2022

Mục lục

1 Giới thiệu **1**

2 Thuật toán ngây thơ **1**

3 Thuật toán Euclid **2**

4 Thuật toán Euclid mở rộng (Extended Euclid Algorithm) **2**

5 Một số bài tập **3**

1 Giới thiệu

Ước chung lớn nhất (GCD, viết tắt của từ Greatest Common Divisor) của hai hay nhiều số là số nguyên dương lớn nhất mà là ước chung (common divisor) của tất cả các số đó. (Ví dụ: GCD của 6 và 10 là 2 vì 2 là số nguyên dương lớn nhất mà là ước chung của 6 và 10.

2 Thuật toán ngây thơ

Ta có thể duyệt tất cả các số từ $min(A, B)$ đến 1 và kiểm tra xem số đang xét có phải là ước của của A và B hay không. Nếu đúng như vậy thì số đang xét sẽ là GCD của A và B.

```
int gcd(int A, int B) {
    for (int i = min(A, B); i > 0; --i)
        if (A % i == 0 && B % i == 0)
            return i;
}
```

Độ phức tạp của thuật toán: ***O(min(A,B))***.

3 Thuật toán Euclid

Thuật toán Euclid dựa trên tính chất sau của ước chung lớn nhất $GCD(A, B) = GCD(B, A\%B)$. Thuật toán sẽ quy nạp cho đến khi $A\%B = 0$.

```
int gcd(int A, int B) {
    if (B == 0) return A;
    else return gcd(B, A % B);
}
```

Ví dụ:
Giả sử A=16, B=10.
$GCD(16, 10) = GCD(10, 16\%10) = GCD(10, 6)$
$GCD(10, 6) = GCD(6, 10\%6) = GCD(6, 4)$
$GCD(6, 4) = GCD(4, 6\%4) = GCD(4, 2)$
$GCD(4, 2) = GCD(2, 4\%2) = GCD(2, 0)$
Vì $B = 0$ nên $GCD(2, 0)$ sẽ trả về giá trị 2.
Độ phức tạp của thuật toán: ***O(log max(A,B))***.

4 Thuật toán Euclid mở rộng (Extended Euclid Algorithm)

Đây là một thuật toán mở rộng của thuật toán Euclid ở trên. $GCD(A, B)$ có một tính chất rất đặc biệt: Nó luôn có thể được biểu diễn dưới dạng phương trình $Ax + By = GCD(A, B)$.
Thuật toán sẽ cho ta biết một cặp giá trị $(x; y)$ thỏa mãn phương trình này và nhờ đó giúp ta tính Modular Multiplicative Inverse. Giá trị của x và y bằng không hoặc âm. Chương trình sau đọc hai số A và B và in ra $GCD(A, B)$ cũng như một cặp số $(x; y)$ thỏa mãn phương trình trên.

```
int d, x, y;
void extendedEuclid(int A, int B) {
    if (B == 0) {
        d = A;
        x = 1;
        y = 0;
    }
    else {
        extendedEuclid(B, A%B);
        int temp = x;
        x = y;
        y = temp - (A/B)*y;
    }
}

int main() {
    int A, B;
    cin >> A >> B;
    extendedEuclid(A, B);
    cout << "gcd(" << A << ", " << B << ") = " << d << endl;
    cout << "x, y: " << x <<  ", " << y << endl;
```

```
    return 0;
}
```

Input:

```
    A = 16
    B = 10
```

Output:

```
    gcd(16, 10) = 2
    x, y: 2, -3
```

Ban đầu, thuật toán Euclid mở rộng sẽ chạy như thuật toán Euclid cho đến khi ta có $GCD(A, B)$ hoặc cho đến khi B bằng 0 và khi đó thuật toán sẽ đặt $x = 1$ và $y = 0$. Vì $B = 0$ và $GCD(A, B)$ là A trong thời điểm hiện tại nên phương trình $Ax + By = 0$ trở thành $A.1 + 0.0 = A$.
Giá trị của các biến d,x,y trong hàm extendedEuclid() sẽ lần lượt trở thành:
$d = 2, x = 1, y = 0$
$d = 2, x = 0, y = 1 - (4/2).0 = 1$
$d = 2, x = 1, y = 0 - (6/4).1 = -1$
$d = 2, x = -1, y = 1 - (10/6).(-1) = 2$
$d = 2, x = 2, y = -1 - (16/10).2 = -3$
Độ phức tạp của thuật toán: ***O(log max(A,B))***.

5 Một số bài tập

Codechef - GCD and LCM

UVA - Gift Dilemma