Алгоритм Евкліда Найбільший спільний дільник

СТУДЕНТ ГРУПИ ФВЕ
КІРГЕТОВ ДАНІІЛ

НСД

Найбільший спільний дільник — найбільше число, на яке можуть поділитись два довільних числа без залишка.

Алгоритм Евкліда для цілих чисел

$$a = b q_0 + r_1$$

$$b = r_1 q_1 + r_2$$

$$r_1 = r_2 q_2 + r_3$$

....

$$r_{k-2} = r_{k-1} q_{k-1} + r_k$$

$$r_{k-1} = r_k q_k + 0$$

```cpp
#include<iostream>

using namespace std;

int gcd(int a, int b) {
    while (a > 0 && b > 0) {
        if (a > b) {
            a %= b;
        }
        else {
            b %= a;
        }
    }
    return a + b;
}

int main() {
    int frst, secnd;
    cin >> frst >> secnd;

    cout << gcd(frst, secnd);

    return 0;
}
```

Геометричний принцип алгоритма Евкліда

Нехай маємо 2 відрізка різної довжини. Віднімаємо від більшого менший, доки перший не стане меншим за другий. Повторюємо, доки не залишаться 2 однакових відрізки. Їх довжина буде найбільшою спільною мірою. Інакше ці відрізки будуть несумірними.

```cpp
#include<iostream>

using namespace std;

int gcd(int a, int b) {
    while (a != b) {
        while (a > b) {
            a -= b;
        }
        while (b > a) {
            b -= a;
        }
    }
    return a;
}

int main() {
    int frst, secnd;
    cin >> frst >> secnd;

    cout << gcd(frst, secnd);

    return 0;
}
```

Бінарний алгоритм Евкліда

1. НСД(0, n) = n; НСД(m, 0) = m; НСД(n, n) = n;
2. НСД(1, n) = 1; НСД(m, 1) = 1;
3. Якщо m, n парні, то НСД(m, n) = 2 * НСД(m/2, n/2);
4. Якщо m парне, n непарне, то НСД(m, n) = НСД(m/2, n);
5. Якщо n парне, m непарне, то НСД(m, n) = НСД(m, n/2);
6. Якщо m, n непарні и $n > m$, то НСД(m, n) = НСД((n-m)/2, m);
7. Якщо m, n непарні и $n < m$, то НСД(m, n) = НСД((m-n)/2, n);

```
int gcd(int a, int b) {
    if (a == 0)
        return b;
    if (b == 0)
        return a;
    if (a == b)
        return a;
    if (a == 1 || b == 1)
        return 1;
    if (a % 2 == 0 && b % 2 == 0)
        return 2 * gcd(a / 2, b / 2);
    if (a % 2 == 0 && b % 2 != 0)
        return gcd(a / 2L, b);
    if (a % 2 != 0 && b % 2 == 0)
        return gcd(a, b / 2);
    if (a < b)
        return gcd((b - a) / 2, a);
    else
        return gcd((a - b) / 2, b);
}
```

Бінарний алгоритм з використанням побітових операцій

x >>= n — побітовий зсув вправо. Зменшує число на n бітів зправа і еквівалентний діленню на 2 з округленням у меншу сторону

x = 12 (1100); x>>=1 —> x=3(11)

x <<= n — побітовий зсув вліво. Еквівалентний множенню на 2

x = 12 (1100); x<<=1 —> x=24(1 1000)

& — побітове І. Порівнюються кожен біт чисел і повертається число, кожний біт якого є результатом операції І

45(10 1101) & 12(1100) = 12 (00 1100)

| — побітове АБО. Порівнюються кожен біт чисел і повертається число, кожний біт якого є результатом операції АБО

10 (1010) | 5 (0101) = 15 (1111)

```
int gcd(int a, int b) {
    int nod = 1;
    int tmp;
    if (a == 0)
        return b;
    if (b == 0)
        return a;
    if (a == b)
        return a;
    if (a == 1 || b == 1)
        return 1;
    while (a != 0 && b != 0) {
        if (((a & 1) | (b & 1)) == 0) {
            nod <<= 1;
            a >>= 1;
            b >>= 1;
            continue;
        }
        if (((a & 1) == 0) && (b & 1)) {
            a >>= 1;
            continue;
        }
        if ((a & 1) && ((b & 1) == 0)) {
            b >>= 1;
            continue;
        }
        if (a > b) {
            tmp = a;
            a = b;
            b = tmp;
        }
        tmp = a;
        a = (b - a) >> 1;
        b = tmp;
    }
    if (a == 0)
        return nod * b;
    else
        return nod * a;
}
```

Еффективність різновидів алгоритму Евкліда

Середня швидкість виконнання алгоритму для випадкових пар чисел

```
Среднее время для 500 пар случайных чисел:
01 перебор от произвольного числа       :   0,3727 сек.
02 перебор от минимального числа        :   0,2126 сек.
03 с разложением на делители            :   0,0051 сек.
04 алгоритм Евклида рекурсивный         :   0,0053 сек.
05 алгоритм Евклида итерационный        :   0,0007 сек.
06 бинарный алгоритм рекурсивный        :   0,0024 сек.
07 бинарный алгоритм итерационный       :   0,0006 сек.
08 бинарный алгоритм итерац. со сдвигом :   0,0006 сек.

C:\Users\danie\source\repos\gcd\Debug\gcd.exe (process 47816
Press any key to close this window . . .
```

Дякую за увагу