НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Алгоритмы основанные на теории чисел

Беркунский Е.Ю., кафедра ИУСТ, НУК
eugeny.berkunsky@gmail.com
http://www.berkut.mk.ua

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

НОД, НОК

Даны два натуральных числа A и B.

- Найти их наибольший общий делитель (НОД)
- Найти их наименьшее общее кратное (НОК)

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

НОД, НОК

Даны два натуральных числа A и B.

- Найти их наибольший общий делитель (НОД)
- Найти их наименьшее общее кратное (НОК)

Начнем с НОД:

Как его искать?

Какие свойства НОД?

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

НОД, НОК

Как будем искать НОД?

Какие свойства НОД?

1) НОД(a,a) = a
2) НОД(a,0) = a
3) НОД(a,b) ≤ min(a,b), если a≠0, b≠0
4) НОД(a*d,b*d) = d*НОД(a,b)

НОД – алгоритм Евклида

Если A = a1*d и B = b1*d и A>B,

то C = A – B, C = c1*d

```
while (a!=b) {
    if (a>b) a -= b;
    else b -= a;
}
```

НОД – алгоритм Евклида

Если A = a1*d и B = b1*d и A>B,

то C = A – B, C = c1*d

```
while (a!=b) {
    if (a>b) a -= b;
    else b -= a;
}
```

НОД – алгоритм Евклида

Если A = a1*d и B = b1*d и A>B,

то C = A – B, C = c1*d

```
while (a!=b) {
    if (a>b) a -= b;
    else b -= a;
}
```

Как это можно оптимизировать?

Улучшенный алгоритм Евклида

```
while (a!=0 && b!=0) {
 if (a>b) a %= b; else b %= a;
}
return a+b;
```

А как насчет НОК?

```
while (a!=0 && b!=0) {
 if (a>b) a %= b; else b %= a;
}
return a+b;
```

А как найти НОК?

НОК (a,b) = a*b/НОД(a,b)

Проблемы с переполнением..?

Расширенный алгоритм Евклида

Найти НОД(a,b)=d и целые числа x и y, такие что ax + by = d

НА ВХОДЕ: два неотрицательных числа a и b: a>=b

НА ВЫХОДЕ: d=НОД(a,b) и целые x,y: ax + by = d.

1. Если b=0 положить d=a, x=1, y=0 и возвратить (d,x,y)
2. Положить x2=1, x1=0, y2=0, y1=1
3. Пока b>0
    1. q=[a/b], r=a-qb, x=x2-qx1 , y=y2-qy1
    2. a=b, b=r, x2=x1 , x1=x, y2=y1 , y1=y4.
4. d=a, x=x2 , y=y2 и возвратить (d,x,y)

Быстрая степень

- Вычислить a^n с использованием минимального количества арифметических операций.

Прямое вычисление:

```
result = 1;
for (int i=0; i<n; i++) result*=a;
```

НЕ ОПТИМАЛЬНО!

Быстрая степень

- Вычислить a^n с использованием минимального количества арифметических операций.

```
result = 1;
while (n>0) {
    if (n & 1 == 1) {
        result *= a;
        n--;
    } else {
        a = a*a;
        n /= 2;
    }
}
```

Проверка простоты числа

Простое число, это целое число, которое делится только на 1 и на само себя.

Примечание: Число 1 – простым не считается.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Простые числа

Дано целое число N (N>0)

1. Вывести первые N простых чисел.
2. Вывести все простые числа, не превышающие N

Простые числа

Дано целое число N (N>0)

2. Вывести все простые числа, не превышающие N

Решето Эратосфена

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Решето Эратосфена

1. Расположим числа от 2 до n в таблицу (*решето*) и зачеркнём сначала чётные числа, следующие после двойки.
2. Двойку обведём.
3. Затем найдём следующее после двойки незачёркнутое число (это будет тройка), обведём его, и зачеркнём каждое третье число после тройки (начиная с шести).
4. После этого снова найдём первое после тройки незачёркнутое число (пятёрку), и после него зачеркнём каждое пятое (начиная с десяти).
5. Будем повторять подобные действия, пока в таблице не останутся либо обведённые, либо зачёркнутые числа.
6. Обведённые числа будут в точности простыми