Лабораторная работа 4

Вычисление наибольшего общего делителя

Греков Максим Сергеевич

2022 Москва

RUDN University, Moscow, Russian Federation

Цель работы

Цель работы

- Ознакомиться с алгоритмами вычисления наибольшего общего делителя.
- Реализовать рассмотренные алгоритмы программно.

Описание

Описание

Наибольшим общим делителем двух чисел a и b называется наибольшее число, на которое a и b делятся без остатка.

Для записи может использоваться аббревиатура *НОД*. Например:

- НОД(12345, 24690) = 12345
- НОД(12345, 54321) = 3
- НОД (12345, 12541) = 1

Алгоритмы

Алгоритмы

В данной работе будут рассматриваться следубщие алгоритмы вычисления наибольшего общего делителя:

- Алгоритм Евклида
- Бинарный алгоритм Евклида
- Расширенный алгоритм Евклида
- Расширенный бинарный алгоритм Евклида

Алгоритм Евклида

Для вычисления наибольшего общего делителя двух целых чисел применяется способ повторного деления с остатком, называемый алгоритмом Евклида (рис. 1), а также дополненную версию, называемую расширенным алгоритмом Евклида (рис. 3)

1. Алгоритм Евклида.

Вход. Целые числа a, b; $0 < b \leq a$.

Выход. $d =$ НОД(a, b).

1. Положить $r_0 \leftarrow a, r_1 \leftarrow b, i \leftarrow 1$.
2. Найти остаток r_{i+1} от деления r_{i-1} на r_i.
3. Если $r_{i+1} = 0$, то положить $d \leftarrow r_i$. В противном случае положить $i \leftarrow i + 1$ и вернуться на шаг 2.
4. Результат: d.

Figure 1: Алгоритм Евклида

Бинарный алгоритм Евклида

Бинарный алгоритм Евклида (рис. 2) и его дополненная версия под названием расширенный алгоритм Евклида (рис. 4) являются более быстрыми при реализации на компьютере, поскольку используют двоичное представление чисел a и b.

Бинарный алгоритм Евклида

2. Бинарный алгоритм Евклида.

Вход. Целые числа a, b; $0 < b \leq a$.

Выход. $d =$ НОД(a, b).

1. Положить $g \leftarrow 1$.
2. Пока оба числа a и b четные, выполнять $a \leftarrow \frac{a}{2}$, $b \leftarrow \frac{b}{2}$, $g \leftarrow 2g$ до получения хотя бы одного нечетного значения a или b.
3. Положить $u \leftarrow a, v \leftarrow b$.
4. Пока $u \neq 0$ выполнять следующие действия:

   4.1. Пока u четное, полагать $u \leftarrow \frac{u}{2}$.

   4.2. Пока v четное, полагать $v \leftarrow \frac{v}{2}$.

   4.3. При $u \geq v$ положить $u \leftarrow u - v$. В противном случае положить $v \leftarrow v - u$.
5. Положить $d \leftarrow gv$.
6. Результат: d

Figure 2: Бинарный алгоритм Евклида

Расширенный алгоритм Евклида

3. Расширенный алгоритм Евклида.

Вход. Целые числа a, b; $0 < b \leq a$.

Выход. $d = \text{НОД}(a, b)$; такие целые числа x, y, что $ax + by = d$.

1. Положить $r_0 \leftarrow a, r_1 \leftarrow b, x_0 \leftarrow 1, x_1 \leftarrow 0, y_0 \leftarrow 0, y_1 \leftarrow 1, i \leftarrow 1$.
2. Разделить с остатком r_{i-1} на r_i: $r_{i-1} = q_i r_i + r_{i+1}$.
3. Если $r_{i+1} = 0$, то положить $d \leftarrow r_i$, $x \leftarrow x_i$, $y \leftarrow y_i$. В противном случае положить $x_{i+1} \leftarrow x_{i-1} - q_i x_i, y_{i+1} \leftarrow y_{i-1} - q_i y_i, i \leftarrow i + 1$ и вернуться на шаг 2.
4. Результат: d, x, y.

Figure 3: Расширенный алгоритм Евклида

Расширенный бинарный алгоритм Евклида

4. Расширенный бинарный алгоритм Евклида.

Вход. Целые числа a, b; $0 < b \le a$.

Выход. $d = \text{НОД}(a, b)$.

1. Положить $g \leftarrow 1$.
2. Пока числа a и b четные, выполнять $a \leftarrow \frac{a}{2}$, $b \leftarrow \frac{b}{2}$, $g \leftarrow 2g$ до получения хотя бы одного нечетного значения a или b.
3. Положить $u \leftarrow a, v \leftarrow b, A \leftarrow 1, B \leftarrow 0, C \leftarrow 0, D \leftarrow 1$.
4. Пока $u \ne 0$ выполнять следующие действия:

   4.1. Пока u четное:

      4.1.1. Положить $u \leftarrow \frac{u}{2}$.

      4.1.2. Если оба числа A и B четные, то положить $A \leftarrow \frac{A}{2}, B \leftarrow \frac{B}{2}$. В противном случае положить $A \leftarrow \frac{A+b}{2}$, $B \leftarrow \frac{B-a}{2}$.

   4.2. Пока v четное:

      4.2.1. Положить $v \leftarrow \frac{v}{2}$.

      4.2.2. Если оба числа C и D четные, то положить $C \leftarrow \frac{C}{2}, D \leftarrow \frac{D}{2}$. В противном случае положить $C \leftarrow \frac{C+b}{2}, D \leftarrow \frac{D-a}{2}$.

   4.3. При $u \ge v$ положить $u \leftarrow u - v, A \leftarrow A - C, B \leftarrow B - D$. В противном случае положить $v \leftarrow v - u, C \leftarrow C - A, D \leftarrow D - B$.

5. Положить $d \leftarrow gv, x \leftarrow C, y \leftarrow D$.
6. Результат: d, x, y.

Figure 4: Расширенный бинарный алгоритм Евклида

Реализация

Алгоритм Евклида

```
def euclid(self, a: int, b: int) -> int:
    r0 = a
    r1 = b
    while r1!=0:
        r0 = r0%r1
        r0, r1 = r1, r0
    return r0
```

Figure 5: Реализация алгоритма Евклида

Бинарный алгоритм Евклида

```python
def binary_euclid(self, a: int, b: int) -> int:
  even = lambda x: not x%2
  g = 1
  while even(a) and even(b):
    a //= 2
    b //= 2
    g *= 2
  u = a
  v = b
  while u!=0:
    while even(u):
      u //= 2
    while even(v):
      v //= 2
    if u>=v:
      u -= v
    else:
      v -= u
  return g*v
```

Figure 6: Реализация бинарного алгоритма Евклида

Расширенный алгоритм Евклида

```python
def extend_euclid(self, a: int, b: int) -> int:
  r0 = a
  r1 = b
  x0 = 1
  x1 = 0
  y0 = 0
  y1 = 1
  i = 1

  while r1!=0:
    q = r0//r1
    r0 = r0%r1
    r0, r1 = r1, r0

    x0 -= q*x1
    x0, x1 = x1, x0

    y0 -= q*y1
    y0, y1 = y1, y0

  return f'{a}*({x0}) + {b}*({y0}) = {r0}'
```

Figure 7: Реализация расширенного алгоритма Евклида

Расширенный бинарный алгоритм Евклида

```python
def extend_binary_euclid(self, a: int, b: int) -> int:
  even = lambda x: not x%2
  g = 1
  a_copy = a
  b_copy = b
  while even(a) and even(b):
    a //= 2
    b //= 2
    g *= 2
  u = a
  v = b
  A = 1
  B = 0
  C = 0
  D = 1
  while u!=0:
    while even(u):
      u //= 2
      if even(A) and even(B):
        A //= 2
        B //= 2
```

Figure 8: Реализация расширенного бинарного алгоритма Евклида (1)

Расширенный бинарный алгоритм Евклида

```
    else:
        A = (A+b) // 2
        B = (B-a) // 2

    while even(v):
        v //= 2
        if even(C) and even(D):
            C //= 2
            D //= 2
        else:
            C = (C+b) // 2
            D = (D-a) // 2
    if u>=v:
        u-=v
        A-=C
        B-=D
    else:
        v-=u
        C-=A
        D-=B
return f'{a_copy}*({C}) + {b_copy}*({D}) = {g*v}'
```

Figure 9: Реализация расширенного бинарного алгоритма Евклида (2)

Результат

method	GCD(12345,24690)	GCD(12345,54321)	GCD(12345,12541)	GCD(140,96)
binary_euclid	12345	3	1	4
euclid	12345	3	1	4
extend_binary_euclid	12345*(12345) + 24690*(-6172) = 12345	12345*(-14490) + 54321*(3293) = 3	12345*(4159) + 12541*(-4094) = 1	140*(11) + 96*(-16) = 4
extend_euclid	12345*(1) + 24690*(0) = 12345	12345*(3617) + 54321*(-822) = 3	12345*(4159) + 12541*(-4094) = 1	140*(11) + 96*(-16) = 4

Figure 10: Результат

Вывод

Вывод

- Ознакомились с алгоритмами вычисления наибольшего общего делителя.
- Реализовали рассмотренные алгоритмы программно.