Алгоритм Евклида

Поиск наибольшего общего делителя (НОД)

Алгоритм Евклида используется для нахождения НОД двух целых чисел (a) и (b), где

$$(0 < b \leq a)$$

Алгоритм Евклида

Пошаговое описание

1. **Инициализация**:
   - Задаем начальные значения:

$$r_0 \leftarrow a, \quad r_1 \leftarrow b$$

2. **Деление с остатком**:
   - Находим остаток от деления:

r_{i-1} на r_i. Обозначим этот остаток как r_{i+1}.

Алгоритм Евклида

3. **Проверка на завершение**:
   - Если (r_{i+1} = 0), то (d ⟵ r_i), где (d) — это НОД двух чисел (a) и (b).
   - Если (r_{i+1} ≠ 0), увеличиваем (i) на 1 и повторяем шаг 2.
4. **Результат**:
   - После завершения алгоритма (d) содержит НОД чисел (a) и (b).

Бинарный алгоритм Евклида

Бинарный алгоритм Евклида — улучшенная версия, которая быстрее выполняется на компьютерах благодаря операциям сдвига для работы с двоичными числами.

Бинарный алгоритм Евклида

Пошаговое описание

1. **Инициализация множителя**:
   - Устанавливаем (g ⟵ 1). Этот множитель (g) будет использоваться для хранения общего множителя 2, если оба числа (a) и (b) четные.
2. **Удаление общих множителей 2**:
   - Пока (a) и (b) четные, делим их на 2 и умножаем (g) на 2:

$$a \leftarrow \frac{a}{2} \qquad b \leftarrow \frac{b}{2} \qquad g \leftarrow 2g$$

3. **Инициализация переменных**:
    - Устанавливаем (u ⟵ a) и (v ⟵ b).
4. **Основной цикл**:
    - Пока (u ≠ 0), выполняем:
        - Если (u) четное: (u ⟵ u / 2)
        - Если (v) четное: (v ⟵ v / 2)
        - Если (u ≥ v): (u ⟵ u - v); иначе (v ⟵ v - u)

Бинарный алгоритм Евклида

Результат

5. **Вычисление НОД**:
   - Когда (u = 0), устанавливаем (d \leftarrow g \times v). Это и будет НОД чисел (a) и (b).
6. **Итог**:
   - Итоговое значение (d) является наибольшим общим делителем чисел (a) и (b).

Расширенный алгоритм Евклида

Поиск НОД и линейных коэффициентов

Расширенный алгоритм Евклид находит НОД чисел (a), (b), а также целые числа (x) и (y), такие что:

$$ax + by = d$$

Расширенный алгоритм Евклида

Пошаговое описание

1. **Инициализация:**

$$r_0 \leftarrow a, \quad r_1 \leftarrow b, \quad x_0 \leftarrow 1, \quad x_1 \leftarrow 0, \quad y_0 \leftarrow 0, \quad y_1 \leftarrow 1, \quad i \leftarrow 1$$

2. **Деление с остатком:**

$$r_{i-1} = q_i \cdot r_i + r_{i+1}$$

где (q_i) — частное, а (r_{i+1}) — остаток.

3. **Проверка на завершение**:
   - Если (r_{i+1} = 0), то:

$$d \leftarrow r_i, \quad x \leftarrow x_i, \quad y \leftarrow y_i$$

   - Иначе обновляем коэффициенты и возвращаемся к шагу 2.

4. **Результат**:
   - После завершения алгоритма:

$$ax + by = d$$

Расширенный бинарный алгоритм Евклида

Улучшенная версия для линейного представления

Расширенный бинарный алгоритм Евклида позволяет находить НОД и коэффициенты (x) и (y) для линейного представления (ax + by = d).

Расширенный бинарный алгоритм Евклида

Пошаговое описание

1. **Инициализация множителя:**

$$g \leftarrow 1$$

2. **Удаление общих множителей 2:**

$$a \leftarrow \frac{a}{2}, \quad b \leftarrow \frac{b}{2}, \quad g \leftarrow 2g$$

Расширенный бинарный алгоритм Евклида

Основной цикл

3. **Начальные значения**:

$$u \leftarrow a, \quad v \leftarrow b, \quad A \leftarrow 1, \quad B \leftarrow 0, \quad C \leftarrow 0, \quad D \leftarrow 1$$

4. **Основной цикл**:
   - Пока (u \neq v), выполняем:
     - Если (u) четное, делим (u) и обновляем (A), (B).
     - Если (v) четное, делим (v) и обновляем (C), (D).

Расширенный бинарный алгоритм Евклида

Вывод результата

5. **Результат**:
   - Устанавливаем:

$$d \leftarrow g \times v, \quad x \leftarrow C, \quad y \leftarrow D$$

   - Итоговое значение (d), (x), и (y) такое, что:

$$ax + by = d$$

Заключение

Алгоритмы Евклида и их расширенные версии эффективны для нахождения НОД и линейных комбинаций. Бинарные алгоритмы оптимизированы для вычислений на компьютере благодаря использованию двоичных операций.