

**Министерство образования и науки Украины
Севастопольский национальный технический
университет**

ФУНДАМЕНТАЛЬНЫЕ АЛГОРИТМЫ
Изучение алгоритмов для нахождения НОД целых чисел

Методические указания
к выполнению лабораторной работы
для студентов, обучающихся по направлению
**6.050101 “Информационные управляющие
системы и технологии”**
очной и заочной форм обучения

**Севастополь
2014**

УДК 510.5

ФУНДАМЕНТАЛЬНЫЕ АЛГОРИТМЫ: методические указания / Сост., Е. Н. Заикина, Ю. В. Коваленко. — Севастополь: Изд-во СевНТУ, 2014— 8 с.

Целью лабораторной является:

1. Познакомиться с алгоритмом вычисления НОД;
2. Изучить основные правила построения алгоритмов вычисления НОД ;
3. Рассмотреть алгоритм Эвклида для нахождения НОД чисел

Методические указания составлены в соответствии с требованиями программы дисциплины «Теория алгоритмов и математическая логика» для студентов направления 6.050101и утверждены на заседании кафедры Информационных систем, протокол № 1 от 29 августа 2014 года.

Допущено учебно-методическим центром СевНТУ в качестве методических указаний.

Рецензент: Скороход Б.А., доктор технических наук, проф. кафедры ТК.

СОДЕРЖАНИЕ

	Стр.
1 ЦЕЛЬ РАБОТЫ	4
2 ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ	4
2.1 АЛГОРИТМ ЕВКЛИДА	4
2.2 БИНАРНЫЙ АЛГОРИТМ	4
2.3 ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ	5
3 ХОД РАБОТЫ	6
4 ВАРИАНТЫ ЗАДАНИЙ	6
5 СОДЕРЖАНИЕ ОТЧЁТА	7
6 КОНТРОЛЬНЫЕ ВОПРОСЫ	7
7 БИБЛИОГРАФИЧЕСКИЙ СПИСОК	7

1. ЦЕЛЬ РАБОТЫ

1. Познакомиться с алгоритмами для вычисления НОД;
2. Изучить основные правила построения алгоритмов вычисления НОД чисел;
3. Рассмотреть алгоритм Эвклида для нахождения НОД чисел.

2. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

Все задачи представленные в лабораторной работе необходимо решать на языке программирования C++.

Вариант задания для учащегося соответствует его порядковому номеру в группе (этот номер можно узнать у преподавателя).

Задачи данной лабораторной работы направлены на введение в теорию алгоритмов.

Для решения представленных задач необходимо выполнить следующие требования:

1. Повторить работу в выбранной среде на языке C++ и основные команды операционной среды.
2. Не следует писать операторы сплошным текстом. Их следует располагать структурировано.
3. Рекомендуются любую программу сопровождать комментариями, поясняющими назначение, структуру и другие особенности программы.

В лабораторной работе приводятся основные определения и факты из элементарной теории чисел.

2.1 Алгоритма Евклида

Пусть $a \in \mathbb{Z}$, $b \in \mathbb{N}$, и необходимо найти $d = \text{НОД}(a, b)$. Производится деление a на b с остатком:

$$a = q_0 b + r_0, 0 \leq r_0 \leq b$$

Если $r_0 > 0$, то b делится на r_0 :

$$b = q_1 r_0 + r_1, 0 \leq r_1 \leq r_0$$

Получается последовательность равенств с убывающими остатками r_j :

$$r_{j-2} = q_j r_{j-1} + r_j, j = 0, 1, 2 \dots (1)$$

где $r_{-2} = a$, $r_{-1} = b$. Если r_k — последний ненулевой остаток, то

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

Тогда справедливо равенство $d = \text{НОД}(a, b) = r_k$. Вычисление последовательности (1) и есть алгоритм Евклида.

2.2 Бинарный алгоритм

На входе алгоритма заданы числа $a, b \in \mathbb{N}$, $a \geq b$. На выходе $d = \text{НОД}(a, b)$.

1 шаг. Присвоить r остаток от деления a на b . Затем $a := b$, $b := r$.

2 шаг. Если $b=0$, то выдать a и закончить работу. Иначе $k:=0$, и затем, пока a и b оба четные, выполнять:

$$\begin{cases} k := k + 1, \\ a := a/2, \\ b := b/2. \end{cases}$$

(после этого 2^k идет в НОД(a, b), оставшаяся часть НОД будет нечетной).

3 шаг. По крайней мере, одно из чисел a и b нечетное. Если, a четное, то делать присвоение $a:=a/2$ до тех пор, пока, a не станет нечетным. То же для b .

4 шаг. a и b оба нечетные. Присвоить $t:=(a-b)/2$. Если $t=0$, то выдать $2^k a$ и закончить работу.

5 шаг. До тех пор, пока t четное, делать $t:=t/2$.

6 шаг. t нечетное, $t \neq 0$. Если $t > 0$, то $a:=t$. Если $t < 0$, то $b:=-t$. Идти на 4 шаг.

Конец алгоритма

2.3 Примеры решения задач лабораторной работы:

Даны два натуральных числа a и b , не равные нулю одновременно. Вычислить НОД (a, b) - наибольший общий делитель a и b .

```
int gcd(int a, int b) {           // функция, возвращающая НОД двух чисел
    int k = min(a, b);           // НОД не может быть больше минимума из них
    for( ; a % k || b % k; k--); // перебрать все числа, пока не найдено
    //нужное
    return k;
}
```

Рассмотрим решение данной задачи с помощью алгоритма Эвклида:

Будем считать, что НОД ($0, 0$) = 0. Тогда НОД (a, b) = НОД ($a-b, b$) = НОД ($a, b-a$); НОД ($a, 0$) = НОД ($0, a$) = a для всех $a, b \geq 0$.

```
int gcd(int a, int b) {
    if(b < a)
        swap(a, b); // b > a
    while(a) {       // пока a != 0, в этом случае b будет являться ответом
        b -= a;      // уменьшить b на a
        if(a > b)
            swap(a, b); // b всегда должно быть больше a
    }
    return b;
}
```

Несложно заметить, что данный алгоритм можно улучшить, заменив $b -= a$ на $b %= a$. Действительно, вычитая постоянно a , в итоге получится остаток от деления на него. В этом случае сложность алгоритма будет $O(\log(\max(a, b)))$. Данную функцию рекомендуется реализовать самим.

Кроме того, благодаря алгоритму Эвклида можно за такую же сложность найти наименьшее общее кратное - НОК(a, b). Очевидно, что НОК(a, b)

$= a*b / \text{НОД}(a, b)$. Стоит также упомянуть, что $a*b$ может превысить стандартный тип данных, что необходимо учитывать.

3 ХОД РАБОТЫ

1. Ознакомиться с теоретическим разделом настоящих методических указаний и повторить соответствующий лекционный материал.

2. Разработать алгоритм нахождения НОД или НОК В соответствии с вариантом задания.

3. Реализовать алгоритм, полученный в п.2, на языке высокого уровня C++.

4. Сделать выводы по работе, оформить отчет, подготовить ответы на контрольные вопросы.

4 ВАРИАНТЫ ЗАДАНИЙ

Вариант №1.

Написать алгоритм Евклида, использующий соотношения $\text{НОД}(a, b) = \text{НОД}(a \bmod b, b)$ при $a \geq b$ $\text{НОД}(a, b) = \text{НОД}(a, b \bmod a)$ при $b \geq a$

Вариант №2.

Написать алгоритм Евклида, использующий соотношения $\text{НОД}(a, b) = \text{НОД}(a, b \bmod a)$ при $b \geq a$

Вариант №3

Даны натуральные a и b , не равные 0 одновременно. Найти $d = \text{НОД}(a, b)$ и такие целые x и y , что $d = a*x + b*y$.

Вариант №4

Даны натуральные a и b , не равные 0 одновременно. Найти $d = \text{НОД}(a, b)$ и такие целые x и y , что $d = a*x + b*y$.

Вариант №5

Даны натуральные числа a и b , не равные 0 одновременно. Найти $\text{НОД}(a, b)$ и $\text{НОК}(a, b)$.

Вариант №6.

Даны натуральные a и b , не равные 0 одновременно. Найти $d = \text{НОД}(a, b)$ и такие целые x и y , что $d = a*x + b/y$.

Вариант №7

Написать вариант алгоритма Евклида, использующий соотношения $\text{НОД}(2*a, 2*b) = 2*\text{НОД}(a, b)$

Вариант №8

Написать вариант алгоритма Евклида, использующий соотношения
 $\text{НОД}(2*a, b) = \text{НОД}(a, b)$ при нечетном b

Вариант №9

Написать бинарный алгоритм, использующий соотношения $\text{НОД}(a, b) = \text{НОД}(a \bmod b, b)$ при $a \geq b$ $\text{НОД}(a, b) = \text{НОД}(a, b \bmod a)$ при $b \geq a$

Вариант №10

Даны натуральные a и b , не равные 0 одновременно. Найти $d = \text{НОД}(a, b)$ и такие целые x и y , что $d = a*x + b*y$, используя бинарный алгоритм.

5 СОДЕРЖАНИЕ ОТЧЁТА

1. Цель работы.
2. Вариант задания.
3. Текст программы, реализующей расчеты по соответствующему варианту.
4. Анализ результатов работы программы в виде таблицы результатов и графиков.
5. Развернутый вывод по работе.

6 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Определение алгоритма.
2. Алгоритм Евклида, применение алгоритма Евклида.
3. Сравнение сложностей алгоритма Евклида и простого поиска НОД двух чисел.
4. Нахождение НОК(a, b) при помощи алгоритма Евклида.
5. При каких a, b алгоритм будет работать быстрее всего? Дольше всего?

7 БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Т. Кормен, Ч. Лейзерсон, Р. Ривест Алгоритмы: построение и анализ. М.: МЦИМО, 2001.-960 с., 263 ил.
2. Гулд Х., Тобочник Я. Компьютерное моделирование в физике: в 2-х частях. Часть 2: Пер.с англ. - М.: Мир, 1990. -400 с., ил.
3. Ульянов М.В., Шептунов М.В. Математическая логика и теория алгоритмов, часть 2: Теория алгоритмов. – М.: МГАПИ, 2003. – 80 с.