

Отчёт по лабораторной работе №4

Алгоритмы вычисления наибольшего общего делителя

Студент: Гонсалес Ананина Луис Антонио, 1032175329

Группа: НФИмд-02-21

Преподаватель: Кулябов Дмитрий Сергеевич,
д-р.ф.-м.н., проф.

Москва 2021

Содержание

1	Цель работы	4
2	Теоретические сведения	5
3	Выполнение работы	7
4	Выводы	10
	Список литературы	11

List of Figures

2.1	Бинарный алгоритм	5
2.2	Расширенный алгоритм	6
3.1	Работа1	7
3.2	Работа2	8
3.3	Работа3	9

1 Цель работы

Цель данной лабораторной работы- изучить теорию и реализовать все рассмотренные алгоритмы программно.

2 Теоретические сведения

Алгоритм Евклида – это алгоритм нахождения наибольшего общего делителя (НОД) пары целых чисел.

Наибольший общий делитель (НОД) – это число, которое делит без остатка два числа и делится само без остатка на любой другой делитель данных двух чисел. Проще говоря, это самое большое число, на которое можно без остатка разделить два числа, для которых ищется НОД[1].

Алгоритм нахождения НОД делением Большее число делим на меньшее. Если делится без остатка, то меньшее число и есть НОД (следует выйти из цикла). Если есть остаток, то большее число заменяем на остаток от деления. Переходим к пункту 1.

Бинарный алгоритм Евклида

Бинарный алгоритм вычисления НОД, как понятно из названия, находит наибольший общий делитель, а именно НОД двух целых чисел. В эффективности данный алгоритм превосходит метод Евклида, что связано с использованием сдвигов, то есть операций деления на степень 2-ки, в нашем случае на 2.

$$\begin{aligned}\text{НОД}(2A, 2B) &= 2\text{НОД}(A, B) \\ \text{НОД}(2A, 2B+1) &= \text{НОД}(A, 2B+1) \\ \text{НОД}(-A, B) &= \text{НОД}(A, B)\end{aligned}$$

Figure 2.1: Бинарный алгоритм

Теперь рассмотрим этапы работы алгоритма. Они основываются на приведенных свойствах наибольшего общего делителя.

- 1) инициализируем переменную k значением 1. Ее задача – подсчитывать «несоразмерность», полученную в результате деления. В то время как A и B сокращаются вдвое, она будет увеличиваться вдвое;
- 2) пока A и B одновременно не равны нулю, выполняем а.если A и B – четные числа, то делим надвое каждое из них: $A \leftarrow A/2$, $B \leftarrow B/2$, а k увеличивать вдвое: $k \leftarrow k \cdot 2$, до тех пор, пока хотя бы одно из чисел A или B не станет нечетным;
 - b. если A – четное, а B – нечетное, то делим A , пока возможно деление без остатка;
 - c. если B – четное, а A – нечетное, то делим B , пока возможно деление без остатка;
 - d. если $A \geq B$, то заменим A разностью A и B , иначе B заменим разностью B и A .
- 3) после выхода из 2-ого пункта, остается вернуть в качестве результата произведение B и k : $\text{НОД}(A, B) = B \cdot k[2]$.

Расширенный алгоритм Евклида

В то время как “обычный” алгоритм Евклида просто находит наибольший общий делитель двух чисел a и b , расширенный алгоритм Евклида находит помимо НОД также коэффициенты x и y такие, что:

$$a \cdot x + b \cdot y = \text{gcd}(a, b).$$

Figure 2.2: Расширенный алгоритм

Т.е. он находит коэффициенты, с помощью которых НОД двух чисел выражается через сами эти числа[3].

3 Выполнение работы

```
In [1]: def alg_euclid(a,b):
        assert 0<b<=a

        r=[a,b]
        while r[-2]%r[-1]!=0:
            r.append(r[-2]%r[-1])
        return r[-1]

In [2]: alg_euclid(14,6)

Out[2]: 2

In [3]: def bin_euclid(a,b):
        assert 0<b<=a
        g=1

        while a%2==0 and b%2==0:
            a=a/2
            b=b/2
            g*=2

        u=a
        v=b
        while u!=0:
            while u%2==0:
                u=u/2
            while v%2==0:
                v=v/2
            if u>=v:
                u=u-v
            else:
                v=v-u

        d=g*v
        return int(d)

bin_euclid(14,6)

Out[3]: 2
```

Figure 3.1: Работа1

```

In [4]: def ext_euclid(a,b):
        assert 0<b<=a

        u=[a,1,0]
        v=[b,0,1]
        while v[0]!=0:
            q=u[0]//v[0]
            t=[u[0]-q*v[0], u[1]-q*v[1],u[2]-q*v[2]]
            u,v=v,t
        return u[0]

ext_euclid(14,6)

```

Out[4]: 2

```

In [8]: def bin_ext_euclid(a,b):
        assert 0<b<=a

        g=1
        while a%2==0 and b%2==0:
            a=a/2
            b=b/2
            g*=2

        u=a
        v=b
        A=1
        B=0
        C=0
        D=1
        while u!=0:
            while u%2==0:
                u=u/2
                if A%2==0 and B%2==0:
                    A=A/2
                    B=B/2
                else:
                    A=(A+b)/2
                    B=(B-a)/2
            while v%2==0:
                v=v/2
                if C%2==0 and D%2==0:
                    C=C/2
                    D=D/2
                else:
                    C=(C+b)/2
                    D=(D-a)/2

```

Figure 3.2: Работа2


```

        D=D//2
    else:
        C=(C+b)//2
        D=(D-a)//2
    if u>=v:
        u=u-v
        A=A-C
        B=B-D
    else:
        v=v-u
        C=(C+b)//2
        D=(D-a)//2
    d=g*v
    x=C
    y=D
    return int(d),x,y

bin_ext_euclid(14,6)[0]

ut[8]: 2

n [ ]:

```

Figure 3.3: Рабoтa 3

4 Выводы

В итоге в данной лабораторной работы я изучил теорию и реализовал все рассмотренные алгоритмы программно.

Список литературы

1. Алгоритм Эвклида [Электронный ресурс]. Википедия, 2021. URL: <https://younglinux.info/algorithm/euclidean>.
2. Бинарный алгоритм Эвклида [Электронный ресурс]. Википедия, 2021. URL: <https://kvodo.ru/binarnyy-algoritm-vyichisleniya-nod.html>.
3. Расширенный алгоритм Эвклида [Электронный ресурс]. Википедия, 2021. URL: http://e-maxx.ru/algo/export_extended_euclid_algorithm.