

1 Задачи работы

Для каждой пары чисел найти наибольший общий делитель и его линейное представление, используя следующие алгоритмы:

1. Расширенный алгоритм Евклида.
2. Расширенный бинарный алгоритм Евклида.
3. Расширенный алгоритм Евклида с «усечёнными» остатками.

Привести все итерации алгоритмов (последовательность остатков и две последовательности коэффициентов линейного представления). При числе итераций больше 20 привести первые 5 и последние 5 элементов каждой последовательности. Итерации должны быть пронумерованы.

При получении различных линейных представлений одного и того же наибольшего общего делителя обосновать корректность полученных результатов. Привести формулу для вычисления линейных представлений.

Привести сравнение быстродействия реализованных алгоритмов и объяснить полученные результаты.

2 Теоретические сведения

Определение НОД:

Целое число $d \neq 0$ называется *наибольшим общим делителем* целых чисел a_1, a_2, \dots, a_k (обозначается $d = \text{НОД}(a_1, a_2, \dots, a_k)$), если выполняются следующие условия:

1. Каждое из чисел a_1, a_2, \dots, a_k делится на d ;
2. Если $d_1 \neq 0$ – другой общий делитель чисел a_1, a_2, \dots, a_k , то d делится на d_1 .

Теорема (о существовании и линейном представлении наибольшего общего делителя):

Для любых целых чисел a_1, a_2, \dots, a_k существует наибольший общий делитель d , и его можно представить в виде линейной комбинации этих чисел:

$$d = c_1 a_1 + c_2 a_2 + \dots + c_k a_k, \text{ где } c_i \in \mathbb{Z}$$

Линейное представление $ax + by = d$ не единственно. Выведем общий вид коэффициентов x и y .

Пусть есть другое представление $ax' + by' = \text{НОД}(a, b)$. Тогда $a(x' - x) + b(y' - y) = 0$. Найдем такие целые числа s, t , для которых $as = bt$. Запишем $a = a_1 \cdot \text{НОД}(a, b)$, где $\text{НОД}(a_1, b_1) = 1$. Отсюда $a_1 s = b_1 t$, и это равенство выполняется для $s = b_1 k, t = a_1 k$, где k – произвольно целое число. Получаем

$$x' = x + \frac{bk}{\text{НОД}(a,b)}, y' = y - \frac{ak}{\text{НОД}(a,b)}$$

для произвольного целого числа k .

2.1 Расширенный алгоритм Евклида

Расширенный алгоритм Евклида находит $\text{НОД}(a, b)$ и его линейное представление, то есть целые числа x и y , для которых $ax + by = d$. Для доказательства корректности алгоритма Евклида понадобятся две леммы.

Лемма 1:

Если числа a и b целые и a делится на b , то $b = \text{НОД}(a, b)$.

Лемма 2:

Для любых целых чисел a, b, c выполняется равенство $\text{НОД}(a + cb, b) = \text{НОД}(a, b)$.

Теорема 1:

Для любых $a, b > 0$ алгоритм Евклида останавливается и выдаваемое им число d является наибольшим общим делителем чисел a и b .

Расширенный алгоритм Евклида находит наибольший общий делитель d чисел a и b и его линейное представление, то есть целые числа x и y , для которых $ax + by = d$, и не требует «возврата».

Сложность алгоритма Евклида равна $O(\log^2 a)$.

Лемма 3:

На каждой итерации расширенного алгоритма Евклида выполняется равенство $ax_i + by_i = r_i$ при $i \geq 0$, где r_i – остатки от деления на каждом шаге алгоритма.

2.2 Расширенный бинарный алгоритм Евклида

Этот вариант алгоритма Евклида оказывается более быстрым при реализации на компьютере, поскольку использует двоичное представление чисел a и b . Бинарный алгоритм Евклида основан на следующих свойствах НОД(a, b) ($0 < b \leq a$):

1. Если оба числа a и b четные, то $\text{НОД}(a, b) = 2 \cdot \text{НОД}(\frac{a}{2}, \frac{b}{2})$;
2. Если число a нечетное, а b – четное, то $\text{НОД}(a, b) = \text{НОД}(a, \frac{b}{2})$;
3. Если оба числа a и b нечетные, $a > b$, то $\text{НОД}(a, b) = \text{НОД}(a - b, b)$;
4. Если $a = b$, то $\text{НОД}(a, b) = a$.

Сложность этого алгоритма – $O(\log^2 a)$. Здесь по аналогии с леммой 3 предыдущего пункта на каждом шаге выполняются соотношения $u = aA + bB$ и $v = aC + bD$. Если все три числа u , A и B четные, то обе части равенства $u = aA + bB$ можно разделить на 2. Если же при четном u хотя бы одно из чисел A , B нечетное, то соотношение $u = aA + bB$ преобразуется в $u = aA + bB + ab - ab = a(A + b) + b(B - a)$. То же справедливо и для чисел v , C , D .

2.3 Расширенный алгоритм Евклида с «усечёнными» остатками

Данная модификация алгоритма Евклида основывается на следующем утверждении: если $r_{i+1} > \frac{r_i}{2}$, то $r_{i+1} = r_i - r_{i+1}$. Сложность данного алгоритма не отличается от сложности расширенного алгоритма Евклида.

3 Ход работы

3.1 Результаты работы программы

Разработанная программа была запущена для всех наборов чисел. Результаты работы программы представлены на Рисунки 1–9.

```

-----EXTENDED EUCLID ALGORITHM-----
Search for GCD((18560553142075070783, 7725867344442366293)):

   r      x      y      q
0  18560553142075070783      1      0      0
1   7725867344442366293      0      1      2
2   3108818453190338197      1     -2      2
3   1508230438061689899     -2      5      2
4    92357577066958399      5     -12     16
5    30509204990355515     -82     197      3
6    829962095891854     251    -603     36
7    630569538248771    -9118    21905      1
8    199392557643083     9369   -22508      3
9    32391865319522    -37225    89429      6
10   5041365725951     232719   -559082      6
11   2143670963816    -1433539    3443921      2
12   754023798319     3099797   -7446924      2
13   635623367178    -7633133    18337769      1
14   118400431141     10732930   -25784693      5
15    43621211473    -61297783    147261234      2
16    31158008195    133328496   -320307161      1
17    12463203278   -194626279    467568395      2
18     6231601639    522581054  -1255443951      2
19           0  -1239788387    2978456297      0

gGCD(18560553142075070783, 7725867344442366293) = 6231601639
x = 522581054
y = -1255443951
[INFO] Check is correct: 18560553142075070783 * 522581054 + 7725867344442366293 * -1255443951 = 6231601639
Time consumed: 0.007609128952026367 sec

```

Рисунок 1 – Результат работы программы для расширенного алгоритма Евклида, набор чисел №1

```

-----EXTENDED EUCLID ALGORITHM-----
Search for GCD((1128260935023581176035370606222706151131, 734804759503482093243321114491967761681)):

   r      x      y      q
0  1128260935023581176035370606222706151131      1      0      0
1   734804759503482093243321114491967761681      0      1      1
2   393456175520099082792049491730738389450      1     -1      1
3   341348583983383010451271622761229372231     -1      2      1
4    52107591536716072340777868969509017219      2     -3      6
<...>
39  10712201970706751852017    -24608845583285961    37785818298741668      2
40   32531632882804611558     6784777906830379    -104177329216210611     32
41   302079448197804282161    -2195737738601858089    3371460353217481220      1
42   23236880630600329397     2263585516508688468    -3475637682433691831     13
43           0    -31622349453214808173    48554750224855475023      0

gGCD(1128260935023581176035370606222706151131, 734804759503482093243321114491967761681) = 23236880630600329397
x = 2263585516508688468
y = -3475637682433691831
[INFO] Check is correct: 1128260935023581176035370606222706151131 * 2263585516508688468 + 734804759503482093243321114491967761681 * -3475637682433691831 = 23236880630600329397
Time consumed: 0.0026073455810546875 sec

```

Рисунок 2 – Результат работы программы для расширенного алгоритма Евклида, набор чисел №2

```

-----EXTENDED EUCLID ALGORITHM-----
Search for GCD((10540471889338057301186033515422097986577273562400307236455973454703489621466811, 17094924008708415355477049334892857510869
      r      x      y      q
0  17094924008708415355477049334892857510869914047576535668110046712802078794701157      1      0      0
1  10540471889338057301186033515422097986577273562400307236455973454703489621466811      0      1      1
2  6554452119370358054291015819470759524292640485176228431654073258098589173234346      1     -1      1
3  3986019769967699246895017695951338462284633077224078804801900196604900448232465     -1      2      1
4  2568432349402658807395998123519421062008007407952149626852173061493688725001881      2     -3      1
<...>
84 39413246293660751966808587686807705123337      108450451006267384938603734871743284799     -175888920166621157656930417578538197846      1
85 20865836273114515747133958187133490947649     -210019810934153433059606379344444757980      340617834375547220166927968505407580119      1
86 1854741002054623621967462949674214175688      318470261940420817998210114216188042779     -516506754542168377823858386083945777965      1
87 2318426252568279527459328687459276771961     -528490072874574251057816493560632800759      857124588917715597990786354589353358084      8
88      0      4546390844937014826460742062701250448851     -7373503465883893161750149222798772642637      0

gGCD(17094924008708415355477049334892857510869914047576535668110046712802078794701157, 10540471889338057301186033515422097986577273562400307236455973454703489621466811)
x = -528490072874574251057816493560632800759
y = 857124588917715597990786354589353358084
[INFO] Check is correct: 17094924008708415355477049334892857510869914047576535668110046712802078794701157 * -528490072874574251057816493560632800759 + 10540471889338057301186033515422097986577273562400307236455973454703489621466811 * 857124588917715597990786354589353358084 = 1
Time consumed: 0.002401590347290039 sec

```

Рисунок 3 – Результат работы программы для расширенного алгоритма Евклида, набор чисел №3

```

-----EXTENDED BINARY EUCLID ALGORITHM-----
Search for GCD((18560553142075070783, 7725867344442366293)):
      u      v      A      B      C      D
0  18560553142075070783      7725867344442366293      1      0      0      1
1  18560553142075070783      7725867344442366293      1      0      0      1
2  10834685797632704490      -      1      -1      -      -
3  5417342898816352245      7725867344442366293      3862933672221183147     -9280276571037535392      0      1
4      -      2308524445626014048      -      -      -3862933672221183147      9280276571037535393
<...>
46 74779219668      -      370658405925988543     -890466368891905657      -      -
47 18694804917      6231601639      2024131437592088709     -4862754877741744110      278257519680954003     -668483323744295470
48 12463203278      -      1745873917911134706     -4194271553997448640      -      -
49 6231601639      6231601639      872936958955567353     -2097135776998724320      278257519680954003     -668483323744295470
50      0      -      594679439274613350     -1428652453254428850      -      -

gGCD(18560553142075070783, 7725867344442366293) = 6231601639
x = 278257519680954003
y = -668483323744295470
[INFO] Check is correct: 18560553142075070783 * 278257519680954003 + 7725867344442366293 * -668483323744295470 = 6231601639
Time consumed: 0.0032219886779785156 sec

```

Рисунок 4 – Результат работы программы для расширенного бинарного алгоритма Евклида, набор чисел №1

```

-----EXTENDED BINARY EUCLID ALGORITHM-----
Search for GCD((1128260935023581176035370606222706151131, 734804759503482093243321114491967761681)):

      u          v          A          B
0  1128260935023581176035370606222706151131  734804759503482093243321114491967761681  1  0
1  1128260935023581176035370606222706151131  734804759503482093243321114491967761681  1  0
2  393456175520099082792049491730738389450  -  1  -1
3  196728087760049541396024745865369194725  734804759503482093243321114491967761681  367402379751741046621660557245983880841  -564130467511790588017685303111353075566
4  -  538076671743432551847296368626598566956  -  -367
<...>
100 139421283783619767382  -  -741167554388324551484957800795689838355  1138030731439960528714433080873222000527  -
101 69710641891808988191  23236880630600329397  -3181397442421229120818343515861038337  48848982081896763406531237325257924698  691633827568282510296273525247058241634  -
102 46473761261200658794  -  -69481522501070373941709186839819279971  1066858733970246751765387216884764332395  -
103 23236880630600329397  23236880630600329397  19994767246389176913114623046524240855  -30701100526667212134991694668970909368  691633827568282510296273525247058241634  -
104 0  -  -67163906032189333383158902200534000779  1031272735235389863289864284890535498329  -

gGCD(1128260935023581176035370606222706151131, 734804759503482093243321114491967761681) = 23236880630600329397
x = 691633827568282510296273525247058241634
y = -1061973835762057075424855979559506407697
[INFO] Check is correct: 1128260935023581176035370606222706151131 * 691633827568282510296273525247058241634 + 734804759503482093243321114491967761681 * -10619738357620570754
Time consumed: 0.003330707550048828 sec

```

Рисунок 5 – Результат работы программы для расширенного бинарного алгоритма Евклида, набор чисел №2

```

-----EXTENDED BINARY EUCLID ALGORITHM-----
Search for GCD((10540471889338057301186033515422097986577273562400307236455973454703489621466811, 17094924008708415355477049334892857510869914047576535668110046712802078794701157)):

      u          v
0  17094924008708415355477049334892857510869914047576535668110046712802078794701157  10540471889338057301186033515422097986577273562400307236455973454703489621466811
1  17094924008708415355477049334892857510869914047576535668110046712802078794701157  10540471889338057301186033515422097986577273562400307236455973454703489621466811
2  6554452119370358054291015819470759524292640485176228431654073258098589173234346  -
3  3277226059685179027145507909735379762146320242588114215827036629049294586617173  10540471889338057301186033515422097986577273562400307236455973454703489621466811  5270
4  -  7263245829652878274040525605686718224430953319812193020628936825654195034849638  -
<...>
192  -  4636852505136559054918657374918553543922  -
193  6955278757704838582377986062377830315883  2318426252568279527459328687459276771961  4224304333380132404616735024772928934928610278379989405598802062160470442482421  -
194  4636852505136559054918657374918553543922  -  5684247799285276952728004637645780724841329803358119001844248979722393241122288  -
195  2318426252568279527459328687459276771961  2318426252568279527459328687459276771961  2842123899642638476364002318822890362420664901679059500922124489861196620561144  -
196  0  -  430206736554778302447527193169574215233384426657189097167571407423119419201011  -

gGCD(17094924008708415355477049334892857510869914047576535668110046712802078794701157, 10540471889338057301186033515422097986577273562400307236455973454703489621466811) =
x = -1459943465905144548111269612872851789912719524978129596245446917561922798639867
y = 2367789873990752853761186109690393342104588768979875352069942369803457409681280
[INFO] Check is correct: 17094924008708415355477049334892857510869914047576535668110046712802078794701157 * -14599434659051445481112696128728517899127195249781295962454469
Time consumed: 0.006136655807495117 sec

```

Рисунок 6 – Результат работы программы для расширенного бинарного алгоритма Евклида, набор чисел №3

```

-----EXTENDED REMAIN EUCLID ALGORITHM-----
Search for GCD((18560553142075070783, 7725867344442366293)):

      r          x          y          q
0  18560553142075070783  1  0  -
1  7725867344442366293  0  1  2
2  3108818453190338197  0  1  2
3  1508230438061689899  1  -2  2
4  92357577066958399  -2  5  16
5  30509204990355515  5  -12  3
6  829962095891854  -82  197  36
7  199392557643083  251  -603  4
8  32391865319522  9369  -22508  6
9  5041365725951  -37225  89429  6
10  2143670963816  232719  -559082  2
11  754023798319  -1433539  3443921  2
12  118400431141  3099797  -7446924  6
13  43621211473  10732930  -25784693  2
14  12463203278  -61297783  147261234  3
15  6231601639  -194626279  467568395  2
16  0  522581054  -1255443951  -

gGCD(18560553142075070783, 7725867344442366293) = 6231601639
x = 522581054
y = -1255443951
[INFO] Check is correct: 18560553142075070783 * 522581054 + 7725867344442366293 * -1255443951 = 6231601639
Time consumed: 0.0057163238525390625 sec

```

Рисунок 7 – Результат работы программы для расширенного алгоритма Евклида с "усеченными" остатками, набор чисел №1

Таблица 1 – Сравнение результатов работы программы

Набор чисел (№п\п), категория выходных данных		Значение данных для алгоритма...		
		Расширенного алгоритма Евклида	Расширенного бинарного алгоритма Евклида	Расширенного алгоритма Евклида с «усечёнными» остатками
1	НОД(a, b)	6231601639	6231601639	6231601639
	x	522581054	278257519680954003	522581054
	y	-1255443951	-668483323744295470	- 1255443951
	Проверка	Пройдена	Пройдена	Пройдена
	Число итераций	19	50	16
2	НОД(a, b)	2323688063060032 9397	232368806306003293 97	2323688063060032 9397
	x	2263585516508688 468	691633827568282510 296273525247058241 634	2263585516508688 468
	y	- 3475637682433691 831	- 106197383576205707 542485597955950640 7697	- 3475637682433691 831
	Проверка	Пройдена	Пройдена	Пройдена
	Число итераций	43	104	29
3	НОД(a, b)	2318426252568279 5274593286874592 76771961	231842625256827952 745932868745927677 1961	2318426252568279 5274593286874592 76771961
	x	- 5284900728745742	- 145994346590514454	- 5284900728745742

		5105781649356063 2800759	811126961287285178 991271952497812959 624544691756192279 8639867	5105781649356063 2800759
	у	8571245889177155 9799078635458935 3358084	236778987399075285 376118610969039334 210458876897987535 206994236980345740 9681280	8571245889177155 9799078635458935 3358084
	Проверка	Пройдена	Пройдена	Пройдена
	Число итераций	88	196	60

Измерения времени работы алгоритмов отражены в Таблица 2. Результаты при первом запуске программы показали противоречивые данные – простой расширенный алгоритм на двух наборах данных оказался быстрее своих улучшенных версий. Было выдвинуто предположение, что причина тому – фиксация результатов каждой итерации и последующий вывод этих данных.

Таблица 2 – Сравнение времени работы алгоритмов

Набор чисел (№п\п)	Время работы алгоритма...(сек)		
	Расширенного алгоритма Евклида	Расширенного бинарного алгоритма Евклида	Расширенного алгоритма Евклида с «усечёнными» остатками
1	0.007609128952026367	0.003221988677978515 6	0.005716323852539062 5
2	0.002607345581054687 5	0.003330707550048828	0.002848386764526367
3	0.002401590347290039	0.006136655807495117	0.00272369384765625

Для того, чтобы проверить предположение, был проведён повторный замер времени «чистой» работы алгоритмов, не включающей запись и вывод данных о промежуточных результатах каждой итерации. Вывод программы показан на рисунках, а сравнение времени – в Таблица 3.

```

-----EXTENDED EUCLID ALGORITHM-----
Search for GCD((18560553142075070783, 7725867344442366293)):

gGCD(18560553142075070783, 7725867344442366293) = 6231601639
x = 522581054
y = -1255443951
[INFO] Check is correct: 18560553142075070783 * 522581054 + 772586734
Time consumed: 3.719329833984375e-05 sec

-----EXTENDED EUCLID ALGORITHM-----
Search for GCD((1128260935023581176035370606222706151131, 73480475950

gGCD(1128260935023581176035370606222706151131, 7348047595034820932433
x = 2263585516508688468
y = -3475637682433691831
[INFO] Check is correct: 1128260935023581176035370606222706151131 * 2
Time consumed: 5.53131103515625e-05 sec

-----EXTENDED EUCLID ALGORITHM-----
Search for GCD((10540471889338057301186033515422097986577273562400307

gGCD(1709492400870841535547704933489285751086991404757653566811004671
x = -528490072874574251057816493560632800759
y = 857124588917715597990786354589353358084
[INFO] Check is correct: 17094924008708415355477049334892857510869914
Time consumed: 0.00011420249938964844 sec

```

Рисунок 10 – Результаты новых замеров времени, расширенный алгоритм

Евклида

```

-----EXTENDED BINARY EUCLID ALGORITHM-----
Search for GCD((18560553142075070783, 7725867344442366293)):

gGCD(18560553142075070783, 7725867344442366293) = 6231601639
x = 278257519680954003
y = -668483323744295470
[INFO] Check is correct: 18560553142075070783 * 278257519680954003 +
Time consumed: 6.794929504394531e-05 sec

-----EXTENDED BINARY EUCLID ALGORITHM-----
Search for GCD((1128260935023581176035370606222706151131, 73480475950

gGCD(1128260935023581176035370606222706151131, 7348047595034820932433
x = 691633827568282510296273525247058241634
y = -106197383576205707542485979559506407697
[INFO] Check is correct: 1128260935023581176035370606222706151131 * 6
Time consumed: 0.00011754035949707031 sec

-----EXTENDED BINARY EUCLID ALGORITHM-----
Search for Gcd((10540471889338057301186033515422097986577273562400307

gGCD(1709492400870841535547704933489285751086991404757653566811004671
x = -1459943465905144548111269612872851789912719524978129596245446917
y = 23677898739907528537611861096903933421045887689798753520699423698
[INFO] Check is correct: 17094924008708415355477049334892857510869914
Time consumed: 0.00028705596923828125 sec

```

Рисунок 11 – Результаты новых замеров времени, расширенный бинарный алгоритм Евклида

```

-----EXTENDED REMAIN EUCLID ALGORITHM-----
Search for GCD((18560553142075070783, 772586734442366293)):

gGCD(18560553142075070783, 772586734442366293) = 6231601639
x = 522581054
y = -1255443951
[INFO] Check is correct: 18560553142075070783 * 522581054 + 772586734
Time consumed: 2.3603439331054688e-05 sec

-----EXTENDED REMAIN EUCLID ALGORITHM-----
Search for GCD((1128260935023581176035370606222706151131, 73480475950

gGCD(1128260935023581176035370606222706151131, 7348047595034820932433
x = 2263585516508688468
y = -3475637682433691831
[INFO] Check is correct: 1128260935023581176035370606222706151131 * 2
Time consumed: 3.981590270996094e-05 sec

-----EXTENDED REMAIN EUCLID ALGORITHM-----
Search for Gcd((10540471889338057301186033515422097986577273562400307

gGCD(1709492400870841535547704933489285751086991404757653566811004671
x = -528490072874574251057816493560632800759
y = 857124588917715597990786354589353358084
[INFO] Check is correct: 17094924008708415355477049334892857510869914
Time consumed: 7.2479248046875e-05 sec

```

Рисунок 12 – Результаты новых замеров времени, расширенный алгоритм Евклида с "усеченными" остатками

Таблица 3 – Сравнение времени "чистой" работы алгоритмов

Набор чисел (№п\п)	Время работы алгоритма...(сек)		
	Расширенного алгоритма Евклида	Расширенного бинарного алгоритма Евклида	Расширенного алгоритма Евклида с «усечёнными» остатками
1	3.719329833984375e-05	6.794929504394531e-05	2.3603439331054688e-05
2	5.53131103515625e-05	0.00011754035949707031	3.981590270996094e-05
3	0.00011420249938964844	0.00028705596923828125	7.2479248046875e-05

Исходя из полученных данных, можно сделать вывод, что при реализации на языке Python наиболее эффективным показывает себя расширенный алгоритм

Евклида с «усеченными» остатками. Это объясняется тем, что по сравнению с другими алгоритмами он выполняет гораздо меньше итераций, и преимущество бинарного алгоритма в скорости вычисления на каждой итерации не настолько велико, чтобы компенсировать большую разницу в числе итераций.

4 Выводы

В ходе выполнения лабораторной работы были реализованы три различных алгоритма для нахождения наибольшего общего делителя (НОД) двух чисел и их линейного представления. При запуске программы было показано на практике, что линейное представление НОД не единственно и зависит от последовательности действий при нахождении коэффициентов. Результаты измерения времени работы программы при использовании различных алгоритмов показали, что на полученных наборах входных данных наиболее эффективным оказался расширенный алгоритм Евклида с «усеченными» остатками по причине малого числа итераций в сравнении с другими алгоритмами.