

Занятие №4. НОД(GCD) и НОК(LCM)

Задачи стр. 5	Подсказки стр. 11	Разборы стр. 12	Справочник стр. 14
---------------	-------------------	-----------------	--------------------

Математикам расшифровывать аббревиатуры в названии не надо, конечно. Для остальных скажем, что это «Наибольший общий делитель» (Greatest Common Divisor) и «Наименьшее общее кратное» (Least Common Multiple).

Проблема, которая обсуждается на этом занятии – это проблема несоответствия.

Известный с детства, простой и понятный для человека алгоритм поиска наибольшего общего делителя (разложить оба числа на простые и вычислить произведение общих простых) довольно сложно реализуется на компьютере.

А трудоемкий – много вычислений, но очень кратко записываемый на языке программирования алгоритм, непонятно почему работает. Попробуем разобраться!

Даны два неотрицательных числа, причем хотя бы одно из них не равно нулю. Будем уменьшать большее число на меньшее (в случае равенства просто любое из них на другое) до тех пор, пока одно не станет равно 0. Тогда ненулевое число будет ответом – наибольшим общим делителем.

Этот алгоритм назван в честь Евклида, который первым изложил его в своих знаменитых «Началах».

Например, возьмем числа 15 и 20. Получаем числа 15 и 5, потом 10 и 5, 5 и 5, наконец 0 и 5. 5 – ответ. Причем верный! По сравнению со «школьным» алгоритмом такая алгоритмическая простота кажется чудом! Но нужно, конечно, доказать, что он работает верно.

Во-первых, легко показать, что алгоритм всегда закончит работу. Попробуйте это сделать самостоятельно. Проверьте свой ответ по Подсказке 4.1.

Теперь посмотрим на пары (a, b) и $(a - b, b)$ (для определенности положим, что $a \geq b$). Оказывается, что у них одни и те же делители! Это доказать несколько сложнее, но можно:

Очевидно, что если $a : d$, и $b : d$, то $(a - b) : d$. (Значок «:» означает «делится»). Математики скучают, а программистам, желающим это строго доказать, подсказка: «число делится на d » значит оно представимо в виде kd .

Но так же аналогично, если $(a - b) : d$ и $b : d$, то и $a : d$

Значит, набор делителей у первоначальных чисел и полученных в результате алгоритма совпадает! У чисел 0 и D наибольший делитель D . А значит, D — это ответ и для исходных чисел.

Алгоритм работает, но слишком долго. Представим, что у нас числа 2 и 1 000 000 000. 500 миллионов вычитаний! «Умереть можно», скажут математики, «Time Limit...» грустно вздохнут программисты. Но ведь что такое много вычитаний из большего числа меньшего? Это нахождение остатка от деления! Действительно, нахождение остатка a на b , это получение числа $0 \leq r < b$, такого что $a = kb + r$, то есть вычитание k раз из a числа b , до тех пор пока результат не станет меньше b !

Можно записывать алгоритм. Оформим его в виде функции. Программисты заметят, что стоит использовать слово `static` в определении, потому что мы будем эту функцию вызывать, скорее всего, без создания объекта класса (например, из `main`).

```
static int hugeGcd(int a, int b)
{
    int res = 0;
    while(a != 0 && b != 0)
    {
        if (a > b) a = a % b;
        else b = b % a;
    }
    if (a > 0) res = a;
    else res = b;
    return res;
}
```

В принципе, это работает. Но функция получилась реально huge, просто огромной. Это потребитель рекламы должен «позволить себе немного лишнего». Мы же постараемся все лишнее наоборот убрать. Математические фокусы закончились, начинаются программистские!

Например, заметим, что одно из чисел будет равно нулю по условию окончания алгоритма, а другое будет ответом. Значит можно выйти без всяких «если» и вернуть сумму! Сразу после цикла одной строкой написать `return a + b;` и вовсе обойтись без переменной `res`.

Понятно, что числа меняются постоянно местами. Мы же ищем остатки, а они получаются меньше чем делимое. Значит можно войти в цикл и постоянно их - `a` и `b` - менять местами. Но ведь для этого нужно быть уверенным, что первое число - `a` - больше второго. Оказывается это необязательно!

```
while (b != 0)
{
    int tmp = a % b;
    a = b;
```

```
    b = tmp;  
}
```

Если оформить цикл так, то – если **b** равно 0 – выходим с ответом **a** – все верно! Если нет, входим в цикл и тут либо **a** сразу не меньше **b** (и остается большим до конца цикла) либо, если это не так, то они - **a** и **b** - просто поменяются местами после первого прохода! Посмотрите внимательно и обязательно постарайтесь понять, почему это происходит.

Приведем полный, вернее сильно похудевший вариант:

Функция вычисления НОД

```
static int gcd(int a, int b)  
{  
    while (b != 0)  
    {  
        int tmp = a % b;  
        a = b;  
        b = tmp;  
    }  
    return a;  
}
```

С НОК после разработки НОД значительно проще. Для того чтобы вычислить НОК надо произведение чисел поделить на их НОД (**свойство 8**, см. ниже). Попробуйте самостоятельно доказать строго, а мы запишем этот факт в виде программы. Следует учесть, что лучше, чтобы не было переполнения, сначала делить на НОД, а потом умножать на **b**.

Функция вычисления НОК

```
static int lcm(int a, int b)  
{  
    return a / gcd(a, b) * b;  
}
```

Естественно, для того, чтобы ей пользоваться, нужно написать в программе еще и функцию gcd.

Немного теории

Приведем несколько свойств НОД и НОК.

Они могут пригодиться при решении задач не только этого, но и следующих занятий.

Свойство 1. $\text{НОД}(a, b, c) = \text{НОД}(\text{НОД}(a, b), c)$

Свойство 2. $\text{НОК}(a, b, c) = \text{НОК}(\text{НОК}(a, b), c)$

Свойство 3. $\text{НОД}(ac, bc) = c \cdot \text{НОД}(a, b)$

Свойство 4. $\text{НОК}(ac, bc) = c \cdot \text{НОК}(a, b)$

Свойство 5. $\text{НОД}(n, n+1, n+2) = 1$

Свойство 6. Числа $\text{НОД}(a, b)$ и $b/\text{НОД}(a, b)$ взаимно простые.

Свойство 7. $\text{НОД}(n, n+k) = \text{НОД}(n, k)$

Свойство 8. $\text{НОК}(a, b) = ab/\text{НОД}(a, b)$

Свойство 9. Для любых целых a, b из которых хотя бы одно не нуль, существуют такие целые числа x, y , что выполняется соотношение: $\text{НОД}(a, b) = x \cdot a + y \cdot b$. (соотношение Безу)

Теория закончена. К практике. Конечно, мы теперь сможем написать программу, решающую задачу «Сокращение дроби». Воспользуемся **Свойством 6**. Для решения просто вычислим НОД числителя и знаменателя, разделим на него и числитель и знаменатель и выведем их. Только надо учесть, что числитель может быть отрицательным. При вычислении НОД просто возьмем его модуль. Это удобнее всего сделать функцией `abs` класса `Math`. Приведем текст почти всей программы. Собственно решение записывается в две строки:

```
int d = gcd(Math.abs(a), b);  
System.out.println(a / d + " " + b / d);
```

Но оказывается, что есть много других задач, при решении которых очень сложно понять, как применять эти алгоритмы и почему это применение работает. Часто это бывает сложнее, чем понять, как и почему работает алгоритм Евклида!

Рассмотрим несколько примеров из предлагаемых на практике этого Занятия задач.

Немного о задачах

Шестеренки из одноименной задачи имеют по **K** и **N** зубцов соответственно. Нужно прокрутить их так, чтобы система вернулась в исходное положение. Через сколько зубцов это произойдет первый раз?

Обозначим необходимое число переменной **m**. Чтобы первая шестеренка вернулась в исходное состояние, **m** должно делиться на **K**. Чтобы вторая - **m** должно делиться на **N**. Из этого следует, что **m** это просто НОК чисел **N** и **K**!

Какова вероятность того, что мы возьмем наугад два натуральных числа a и b и полученная дробь $\frac{a}{b}$ будет несократима? То есть они окажутся взаимнопростыми, их НОД будет равен единице.

Больше 50% или меньше?

Попробуйте предположить точный ответ. А потом обратитесь к Подсказке 4.2.

В задаче «Разрезание на квадраты» от полоски бумаги размером $A \times B$ отрезаются квадраты максимального размера до тех пор, пока не получится квадрат. Спрашивается, сколько квадратов получится? (Ограничения на исходный размер бумаги: $1 \leq A, B \leq 10^9$)

Причем здесь алгоритм Евклида? Трудно догадаться, что отрезание – это... шаг медленного алгоритма (с вычитанием)! Таким образом, если поставить счетчик шагов в этот алгоритм, и просто запустить его на числах А и В, то мы сразу получим ответ. Но ограничения... Цикл для миллиарда будет работать очень долго. Ничего страшного! Поставим счетчик в алгоритм с делением, а увеличивать его будем сразу на остаток от деления. Это будет соответствовать отрезанию по большей стороне до тех пор, пока она не станет меньше.

Оформим решение в виде функции.

```
static int countGCD(int a, int b)
{
    int n = 0; // счетчик
    while (b != 0)
    {
        int tmp = a % b;
        n += a / b; // увеличиваем счетчик
        a = b;
        b = tmp;
    }
    return n; // ответом на задачу является число n
}
```

Задачи

Задача А. Сокращение дроби

Источник: Московская командная олимпиада 2005 года

Дана дробь $\frac{a}{b}$. Требуется ее сократить, то есть записать это же число в виде $\frac{c}{d}$, где c — целое число, d — натуральное число и d минимальное возможное.

Формат входных данных

Вводятся два целых числа a и b ($-100 \leq a \leq 100$, $0 < b \leq 100$).

Формат выходных данных

Выведите два числа c и d .

Примеры

Входные данные	Выходные данные
3 6	1 2
-2 5	-2 5

Задача В. Шестеренки

Даны две сцепленные шестеренки. У одной шестеренки N зубцов, у другой – K . Требуется найти, какое минимальное число поворотов на один зубчик требуется сделать, чтобы шестеренки вернулись в исходное состояние.

Формат входных данных

В единственной строке – два натуральных числа N и K , не превосходящих 10 миллионов.

Формат выходных данных

Выведите искомое количество зубчиков. Гарантируется, что оно не более миллиарда.

Примеры

Входные данные	Выходные данные
2 3	6
6 21	42

Задача С. Сложить две дроби

Даны две рациональные дроби: a/b и c/d . Сложите их и результат представьте в виде несократимой дроби m/n .

Формат входных данных

Программа получает на вход 4 натуральных числа a, b, c, d , не превосходящих 100.

Формат выходных данных

Программа должна вывести 2 натуральных числа m и n такие, что $m/n = a/b + c/d$ и дробь m/n – несократима.

Примеры

Входные данные	Выходные данные
1 3 1 3	2 3

Задача D. Разрезание на квадраты

Полоска бумаги имеет размеры $A \times B$. Каждый раз от нее отрезается квадрат максимального размера до тех пор, пока не получится квадрат. Сколько квадратов получится?

Формат входного файла

Программе даны числа A и B ($1 \leq A, B \leq 10^9$).

Формат выходного файла

Требуется вывести количество квадратов.

Примеры

Входные данные	Выходные данные
15 3	5
12 8	3
5 5	1

Задача Е. Дробь

Источник: Командная олимпиада школьников Санкт-Петербурга 2007 года.

Коля учится в третьем классе, сейчас они проходят простые дроби с натуральными числителем и знаменателем. Вчера на уроке Коля узнал, что дробь называется *правильной*, если ее числитель меньше знаменателя, и *несократимой*, если нет равной ей дроби с меньшими натуральными числителем и знаменателем.

Коля очень любит математику, поэтому дома он долго экспериментировал, придумывая и решая разные задачки с правильными несократимыми дробями. Одну из этих задач Коля предлагает решить вам с помощью компьютера.

Найдите наибольшую правильную несократимую дробь, у которой сумма числителя и знаменателя равна n .

Входные данные

Во входном файле записано одно целое число n ($3 \leq n \leq 1000$).

Выходные данные

Выведите в выходной файл числитель и знаменатель искомой дроби.

Примеры

Входные данные	Выходные данные
10	3 7
2 3	11 12

Задача F. Представление чисел

Источник: Всероссийская командная олимпиада школьников 2005 года

Дано натуральное число N . Требуется представить его в виде суммы двух натуральных чисел A и B таких, что НОД (наибольший общий делитель) чисел A и B — максимален.

Формат входных данных

Во входном файле записано натуральное число N ($2 \leq N \leq 10^9$)

Формат выходных данных

В выходной файл выведите два искоемых числа A и B . Если решений несколько, выведите любое из них.

Примеры

Входные данные	Выходные данные
15	5 10
16	8 8

Задача G. Водолей

У исполнителя «Водолей» есть два сосуда, первый объемом A литров, второй объемом B литров, а также кран с водой. Водолей может выполнять следующие операции:

Наполнить сосуд A (обозначается $>A$).

Наполнить сосуд B (обозначается $>B$).

Вылить воду из сосуда A (обозначается $A>$).

Вылить воду из сосуда B (обозначается $B>$).

Перелить воду из сосуда A в сосуд B (обозначается как $A>B$).

Перелить воду из сосуда B в сосуд A (обозначается как $B>A$).

Команда переливания из одного сосуда в другой приводят к тому, что либо первый сосуд полностью опустошается, либо второй сосуд полностью наполняется.

Программа получает на вход три натуральных числа A , B , N , не превосходящих 10^4 .

Вам необходимо вывести алгоритм действий Водолея, который позволяет получить в точности N литров в одном из сосудов, если же такого алгоритма не существует, то программа должна вывести текст Impossible.

Количество операций в алгоритме не должно превышать 10^5 . Гарантируется, что если задача имеет решение, то есть решение, которое содержит не более, чем 10^5 операций.

Ввод	Вывод
------	-------

3	>A
5	A>B
1	>A
	A>B
3	Impossible
5	
100	

Задача Н. Отрезок

Источник: Московская олимпиада, окружной тур 2009 года.

На клетчатой бумаге Петя нарисовал отрезок из точки с координатами (a,b) в точку с координатами (c,d). Через сколько клеток проходит этот отрезок (считается, что отрезок проходит через клетку, если он проходит через ее внутренность, если же он проходит только через вершину или по границе клетки, считается, что он не проходит через клетку).

Входные данные

Вводятся целые числа a, b, c, d. Числа по модулю не превышают 10^9 .

Выходные данные

Выведите одно число — количество клеток, через которые проходит отрезок.

Ввод	Вывод
0 0 6 4	8
3 3 -3 3	0

Задача I. Коробки

Есть две коробки. В первой находится a шаров, во второй b ($0 < a + b < 2147483648$). Шары разрешается перекладывать из одной коробки в другую. Причем перекладывать в любую из коробок можно только столько шаров, сколько в ней находится. Необходимо определить, можно ли все шары сложить в одну коробку.

Входные данные

Каждая строка содержит два целых числа a и b, разделенные пробелом.

Выходные данные

Вывести YES, если можно и NO, если нельзя.

Ввод	Вывод
2 6	YES
8 12	NO
7 9	YES

Задача J. Кинотеатр

Источник: ВКОПШ 2005 года

Марья Ивановна с Марьей Михайловной привели школьников в кинотеатр. Чтобы не было никаких обид, Марья Ивановна построила всех школьников по алфавиту и рассадила их: сначала в первый ряд слева направо, затем во второй слева направо и т.д., заполнив весь зал из n рядов по m кресел. Тут пришла Марья Михайловна и сказала, что ребята сели неправильно – надо пересесть. Она предложила сначала заполнить все первые места от первого ряда к последнему, затем все вторые места и т. д.

Определите, сколько школьников после такой пересадки останется на своем месте.

Например, если $n = 3$ и $m = 3$, то в первом случае дети сядут так:

1 2 3

4 5 6

7 8 9

а во втором – так:

1 4 7

2 5 8

3 6 9

Таким образом, три школьника: 1, 5 и 9 останутся на своих местах.

Формат входных данных

Вводятся два целых числа n и m ($1 \leq n, m \leq 10^9$).

Формат выходных данных

Выведите количество школьников, которые останутся на своих местах.

Примеры

Вход	Выход
3 3	3
2 4	2

Подсказки и решения. 4 Занятие.

4.1

Действительно. Числа не становятся отрицательными (мы же вычитаем из числа меньшее или равное) и в то же время постоянно уменьшаются, потому, что мы не будем вычитать ноль, когда он получится – сразу закончим. Рано или поздно одно из чисел станет равным нулю.

4.2

Вероятность того, что число будет делиться на два $1/2$. Действительно, таких чисел ровно половина, каждое второе. Тогда вероятность того, что оба числа будут четные $1/2 * 1/2 = 1/2^2$. Аналогично оба числа будут делиться на 3 с вероятностью $1/3^2$ и т.д. Таким образом, нам надо посчитать сумму $1/2^2 + 1/3^2 + 1/4^2 + \dots$. Несмотря на то, что ряд бесконечен, он сходится – имеет конечную сумму. Эта сумма равна $\frac{6}{\pi^2}$! Удивительно, как «геометрическое» число π попало в теорию натуральных чисел. Кстати, вероятность получается больше половины. Несократимых дробей больше!

Разбор. Занятие 4

4А. Сокращение дроби

Разбирается в тексте занятия

4В. Шестеренки

Разбирается в тексте занятия

4С. Сложить две дроби

Складываем по правилам сложения обыкновенных дробей (знаменатель на числитель плюс числитель на знаменатель в числителе, произведение знаменателей в знаменателе). Полученную дробь сокращаем как в задаче А.

4Д. Разрезание на квадраты

Разбирается в тексте занятия

4Е. Дробь

Небольшие ограничения на n позволяют перебрать все пары возможных числителей и соответствующих им знаменателей так, чтобы числитель был меньше знаменателя: дробь должна быть правильной. И в процессе перебора выбрать из несократимых (их НОД обязательно должен быть равен единице) выбрать максимальную (это дробь с наименьшей разностью между знаменателем и числителем) и вывести ее уже в сокращенном виде.

4F. Представление чисел

Пусть мы нашли два таких числа. Тогда, ясно, что их НОД делит N . Ясно, что чем меньше частное такого деления, тем больше НОД. Частное, в свою очередь, также является делителем. Соответственно, все, что нам нужно найти, это самый маленький делитель N (кроме единицы), в этом случае НОД будет максимален. В цикле по i от 2 до корня из N будем искать делитель N . Если делитель находится, выводим два числа N/i , $N-N/i$ и завершаем программу. Если после прохода по циклу делитель не найден, т.е. число простое, просто выводим 1 и $N-1$.

4G. Водолей

Исполнитель «Водолей», пожалуй, один из самых известных. Его работа тесно связана с алгоритмом Евклида. По соотношению Безу (свойство НОД №9) найдутся такие x и y , что $\text{НОД}(A, B) = xA + yB$ (x и y могут быть отрицательными).

Если требуемый объем не кратен $\text{НОД}(A, B)$ – решения нет. Если кратен, то решение есть и мы рано или поздно получим его (по условию это как раз гарантируется).

Возьмем для определенности x положительным, а y отрицательным. Тогда x будет соответствовать количеству «вливаний» в систему объемов A , а y – выливанию объемов

В. (Алгоритм прост: наполняем А, выливаем его в В, как только В наполняется – выливаем В).

Конечно, необходимо учесть случай, когда отмеряемое количество больше объема сосуда, как во втором примере. Это можно проверить сразу и вывести Impossible.

4Н. Отрезок

В худшем случае отрезок может пересечь $h + w - 1$ клеток (h и w высота и ширина прямоугольника). Но отрезок пройдет через $\text{НОД}(h, w) - 1$ угловых точек. Значит, окончательный ответ: $h + w - \text{НОД}(h, w)$.

4I Коробки

Для того, чтобы можно было переложить, должно выполняться равенство $a + b = 2^k \cdot \text{НОД}(a, b)$ для некоторого k . Почему? Рассмотрим последнее перекладывание. В нем перекладывается половина шаров. Далее для того, чтобы получить такую ситуацию (половину в одной коробке, половину в другой необходимо было переложить четверть шаров и т.д).

4J Кинотеатр

Доказательство корректности ответа для этой задачи достаточно громоздко, однако до самого ответа можно догадаться, попробовав нарисовать и посчитать несколько вариантов. Естественно, сам ответ мы здесь не приводим.

Занятие 4.Справочник

НОД и НОК

Функция вычисления НОД

```
static int gcd(int a, int b)
{
    while (b != 0)
    {
        int tmp = a % b;
        a = b;
        b = tmp;
    }
    return a;
}
```

Функция вычисления НОК

```
static int lcm(int a, int b)
{
    return a / gcd(a, b) * b;
}
```

Естественно, для того, чтобы ей пользоваться, нужно написать в программе функцию gcd.