Занятие 18. Алгоритм Евклида

В занятии 5 мы уже касались этого вопроса. Теперь рассмотрим алгоритм Евклида для нахождения наибольшего общего делителя двух натуральных чисел более подробно.

Из математики известно, что всякое целое число представляется единственным способом через натуральное число в форме

(1)

В этом случае называют неполным частным от деления на , а остатком. Если , то говорят, что кратно или что является делителем числа .

Обратите внимание, что неполное частное и остаток в формуле (1) для при ненулевом остатке отличаются от того, что дают операторы Паскаля и . В формуле (1) частное есть наибольшее целое, не превосходящее . При этом остаток будет неотрицательным. Например, для , получим

В Паскале же получится , . Знак остатка там всегда совпадает со знаком делимого.

Найдём наибольший общий делитель положительных целых чисел и . Будем обозначать его gcd(,)[[1]](#footnote-1).

Теорема 1

Если для целых чисел , , , выполняется равенство

,

то совокупность общих делителей чисел и совпадает с совокупностью общих делителей чисел и .

Доказательство.

Действительно, пусть — общий делитель чисел и . Это значит, что существуют такие целые и , для которых и . Тогда или . Последнее равенство доказывает, что является делителем числа . Аналогично доказывается, что всякий общий делитель чисел и является также и делителем числа . Таким образом, множество общих делителей чисел и совпадает с множеством общих делителей чисел и . В частности, совпадают их наибольшие общие делители: gcd(,)=gcd(,).

Теперь для неотрицательных целых чисел можно записать ряд равенств

,

, (2)

,

и т.д. Совокупность остатков получается убывающей … Но все эти остатки положительны и меньше . А последовательность убывающих натуральных чисел, большее из которых меньше , не может состоять больше, чем из чисел. Значит, на каком-то шаге мы получим нулевой остаток:

Таким образом, совокупность общих делителей чисел и совпадает с совокупностью общих делителей чисел и , которая совпадает с совокупностью общих делителей чисел и и т.д., пока не дойдём до пары и 0. Наибольший общий делитель последней пары есть .

Мы доказали, что наибольший общий делитель пары натуральных чисел и равен последнему отличному от нуля остатку в приведённом выше ряду (2).

## Рекурсивный алгоритм

Опираясь на теорему 1, можно построить следующую рекурсивную подпрограмму.

Function gcd(a,b : Int64) : Int64;

Begin

If b>0 Then gcd:=gcd(b,a mod b)

Else gcd:=a

End;

Заметим, что условие выхода из рекурсии будет выполнено обязательно, так как появление нулевого остатка неизбежно.

## Итеративный алгоритм

Этот же алгоритм можно представить в виде цикла. При этом будем опираться на ряд (2).

Function gcd(a,b : Int64) : Int64;



Рисунок 1. Алгоритм Евклида

Эта реализация алгоритма Евклида содержит проверку всего одного условия в отличие от алгоритма, рассмотренного в занятии 5. А это значит, что алгоритм, представленный на рис. 1, работает быстрее. Ведь на вычисление значения каждого логического выражения и на передачу управления нужной ветви в условном операторе тратится некоторое время.

Заметим также, что обе реализации алгоритма Евклида, рассмотренные выше: и рекурсивная, и итеративная — правильно работают не только для натуральных чисел, а и в том случае, если одно из чисел натуральное, а другое равно нулю. Если хотя бы одно из чисел отрицательно, для нахождения наибольшего общего делителя можно использовать следующее свойство:

Вспомните, что модуль числа *x* на языке Паскаль вычисляет функция abs(*x*).

## Расширенный алгоритм Евклида

Попробуем выразить наибольший общий делитель *натуральных* чисел и в виде линейной комбинации этих чисел. То есть мы хотим найти такие числа и , что

(3)

Если , то такое разложение получить легко. Достаточно взять , а значением можно выбрать любое число, например ноль:

(4)

Пусть теперь .

Как мы уже знаем (см. теорему 1), в этом случае .

Примем во внимание, что , где — целая часть от деления на .

Теперь для ищем разложение, аналогичное (3).

Преобразуя последнее равенство, получаем

(5)

Равенства (3) и (5) должны выполняться при любых значениях и . Это заведомо будет иметь место, если положить

(6)

Равенства (4) и (6) позволяют организовать рекурсивное вычисление коэффициентов разложения (3). Более того, мы видим, что коэффициенты и можно сделать целыми числами. Ведь при их вычислении используются только умножение, вычитание и *целочисленное* деление (для вычисления ), а начальные значения и , как следует из (4), можно выбрать целочисленными. При этом числа и не обязательно будут натуральными.

В вычислении наибольшего общего делителя двух натуральных чисел и его представления в виде линейной комбинации (3) исходных чисел описанным выше способом состоит суть расширенного алгоритма Евклида.

Ниже приведён пример программы, реализующей расширенный алгоритм Евклида. В ней ищется наибольший общий делитель двух натуральных чисел и его представление в виде линейной комбинации исходных чисел.

{$mode delphi}

Program gcde;

//Расширенный алгоритм Евклида

Procedure gcd\_ext(a, b : int64; var x, y, gcd : int64);

var x1,y1 : int64;

Begin

If b=0 Then

Begin

gcd:=a;

x:=1;

y:=0;

Exit

End;

gcd\_ext(b, a mod b, x1, y1, gcd);

x := y1;

y := x1 - (a div b) \* y1

End;

Var m,n,x,y,g : int64;

Begin

WriteLn('Введите 2 натуральных числа');

Read(m,n);

If (m<=0) or (n<=0) Then

Begin

WriteLn('Числа должны быть натуральными');

Halt

End;

gcd\_ext(m,n,x,y,g);

WriteLn('gcd(',m,',',n,')=',g);

Write(x,'\*',m);

If y>=0 Then Write('+');

Write(y);

WriteLn('\*',n,'=',g);

End.

# Задача

На седьмой интернет-олимпиаде для школьников сезона 2009–2010 года 22 мая 2010 года Санкт-Петербургский государственный университет ИТМО предлагал следующую задачу.

## Обратные числа

|  |  |
| --- | --- |
| Имя входного файла: | inverse.in |
| Имя выходного файла: | inverse.out |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

Позавчера Рома узнал про новую для него операцию в математике — *взятие по модулю*, которая обозначается , например . Напомним, что , если и (все перечисленные числа целые).

Вчера Рома начал перемножать числа, и заметил, что бывают такие случаи, что (причем не обязательно оба числа равны единице). И это его очень заинтересовало, он даже придумал название этому феномену — числа и взаимно обратны по модулю .

Сегодня утром Рома начал рассматривать простые числа . Он доказал, что в таком случае для любого числа : существует ровно одно , такое что . Однако он не знает, как по числам и найти число . Помогите ему!

#### Формат входного файла

В единственной строчке входного файла заданы числа и (). Число — простое.

#### Формат выходного файла

В выходной файл выведите число , такое, что . Оно также должно удовлетворять неравенству .

#### Примеры

|  |  |
| --- | --- |
| inverse.in | inverse.out |
| 3 7 | 5 |
| 6 23 | 4 |

### Решение

Число — простое. Это значит, что у него всего два делителя: 1 и . Число , следовательно, все его делители тоже меньше . Отсюда делаем вывод, что единственный, он же наибольший, общий делитель чисел и равен единице[[2]](#footnote-2). Пользуясь расширенным алгоритмом Евклида, можно найти такие целые и , что

(7)

При этом выполняется соотношение

(8)

Однако не гарантируется истинность неравенства , хотя ясно, что , иначе равенства (7) и (8) не могли бы выполняться.

Действительно, при из (7) следует , или , что невозможно ввиду ограничения .

Равенство (8) выполняется для любого , где — требуемый ответ задачи, а — некоторое целое число, в том числе и отрицательное. Тогда, вычислив , мы получим . (Здесь операция mod понимается в смысле языка Паскаль.) Число , для которого также истинно (8), будет удовлетворять неравенству . Отсюда видно, что ответом задачи будет , т.е.

(9)

Таким образом, решение задачи сводится к следующей последовательности шагов:

1. пользуясь расширенным алгоритмом Евклида, вычислить , удовлетворяющее формуле (7);
2. вычислить по формуле (9).

# Задания для самостоятельного выполнения

1. Дано целых чисел. Найдите их наибольший общий делитель. Сведите эту задачу к задаче нахождения наибольшего общего делителя двух чисел. Используйте алгоритм функции gcd, представленный на рис. 1. Обязательно ли при вызове функции gcd требовать, чтобы было больше, чем ?
2. Напишите программу решения задачи «Обратные числа». Тесты для её проверки возьмите на сайте авторов задачи <http://neerc.ifmo.ru/school/io/archive/20100522/archive-advanced-20100522.rar>
3. В процедуре gcd\_ext при мы полагали , а . А можно ли было переменной не присваивать вообще никакого значения? Ведь для любого . Проверьте своё предположение, подправив текст процедуры gcd\_ext и выполнив пробные запуски программы при различных входных данных, в том числе достаточно больших.
4. Напишите итеративную (не рекурсивную) реализацию процедуры gcd\_ext.
5. Усовершенствуйте расширенный алгоритм Евклида так, чтобы он работал с целыми, а не только натуральными, числами. Напишите программу. Учтите, что в языке Паскаль операция *mod* для отрицательных чисел может давать отрицательный остаток, что не соответствует равенствам (2).
6. Малая теорема Ферма утверждает, что для любого простого и целого , которое не делится на , справедливо соотношение . Используя эту теорему, попробуйте найти решение задачи «Обратные числа», не использующее алгоритм Евклида. (В теореме Ферма операцию mod будем понимать в смысле вычисления неотрицательного остатка, как в формуле (1). Хотя для данной задачи это не столь важно: все числа в ней неотрицательные.) Вам также может помочь тот факт, что остаток от произведения чисел равен остатку от произведения остатков:

# Литература

1. Виноградов И.М. Основы теории чисел. Издание шестое. — Москва, Ленинград: Государственное издательство технико-теоретической литературы, 1952.
2. Д. Кнут. Искусство программирования для ЭВМ. Том 1. Основные алгоритмы. Издание 3. — Издательский дом «Вильямс», 2005.
3. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ. Второе издание. — Москва, Санкт-Петербург, Киев. Издательство «Вильямс», 2010.
4. Деление с остатком. // <http://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81_%D0%BE%D1%81%D1%82%D0%B0%D1%82%D0%BA%D0%BE%D0%BC>
5. <http://neerc.ifmo.ru/school/io/archive/20100522/problems-basic-20100522.pdf>
6. <http://neerc.ifmo.ru/school/io/archive/20100522/archive-advanced-20100522.rar>
7. Michaël Van Canneyt. Reference guide for Free Pascal, version 2.4.2. Пункт 9.8.1.

1. От английского greatest common divisor. В русской литературе применяют также обозначение НОД(,) [↑](#footnote-ref-1)
2. Такие числа называют взаимно простыми [↑](#footnote-ref-2)