



ИЗ ОПЫТА РАБОТЫ

М.С. Долинский, к.т.н., доцент кафедры математических проблем управления Гомельского государственного университета им. Ф. Скорины

Элементы теории чисел: решето Эратосфена

Введение

Автор много лет занимается обучением программированию школьников разных возрастов и первокурсников математического факультета (специальности: «Программное обеспечение информационных технологий» и «Прикладная математика») [1-21]. Все это время автор занимался созданием материалов для самостоятельного изучения школьниками и студентами, стараясь представить материал в как можно более простой, наглядной и понятной форме. В данной статье приводится пример такого материала для обучения решению задач по информатике на тему «Решето Эратосфена». Такой материал может быть интересен для преподавателей как в качестве иллюстрации методики обучения, так и по содержанию. В то же время, автору представляется, что этот материал может оказаться весьма полезным и интересным и для школьников, и для студентов, занимающихся самообучением. Всем заинтересованным предлагается следующий порядок работы: откладывать статью в сторону и пытаться самостоятельно выполнить предлагаемое задание, первый раз –

после прочтения условия задачи, второй раз – после прочтения указаний к решению.

Решето Эратосфена

Так называют следующий способ получения ряда простых чисел.

Из ряда чисел

2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 ... вычеркивают кратные двум;

4, 6, 8, 10, 12, ... – кратные трем:

6, 9, 12, 15, ... – кратные пяти:

10, 15, 20, 25, 30, ... – кратные семи:

14, 21, 28, 35, 42, 49, ... и т. д.

Таким образом, все составные числа будут просеяны, и останутся только простые числа 2, 3, 5, 7, 11, 13 ... (Энциклопедический словарь Ф.А. Брокгауза и И.А. Ефрона. – С.-Пб.: ЕСБЕ, 1890-1907).

Ниже представлена процедура Resheto, которая вышеописанным способом находит все простые числа в диапазоне от 2 до MaxX.

```
const
  MaxX = 4000000
Procedure Resheto;
var
  i,j : longint;
begin
  for i:=2 to MaxX do x[i]:=1;
  for i:=2 to trunc(sqrt(MaxX)) do
    if x[i]=1
      then begin
        j:=i;
        while (j+i)<=MaxX do
          begin
            j:=j+i;
            x[j]:=0;
          end;
        end;
  end;
```

Пояснения. Мы хотим создать массив x из MaxX элементов. Если число i ($2 \leq i \leq \text{MaxX}$) простое, то $x[i]=1$, а иначе (число i не простое (то есть составное)), $x[i]=0$.

Вначале мы прописываем все элементы числом 1. Затем, в цикле, начиная со второго элемента. Если он не равен 1, вычеркиваем (заносим число 0 на соответствующие позиции j) все элементы, кратные i , то есть $i+i$, $i+i+i$, $i+i+i+i$, ... (то есть $2*i$, $3*i$, $4*i$...) пока этот индекс ($k*i$) не вышел за границу массива MaxX .

Математиками доказано, для того, чтобы получить все простые числа в диапазоне от 2 до N , достаточно в этом алгоритме вычеркивать числа кратные всем простым числам от 2 до корня квадратного из N ($\text{sqrt}(N)$). Поскольку граница цикла `for` в языке программирования Паскаль должна быть целой, используется функция `trunc` – для того, чтобы превратить корень из N в целое число отбрасыванием его дробной части.

После выполнения этой процедуры в массиве x хранятся пометки всех простых чисел от 2 до N . Для того, чтобы узнать, является ли число Z ($2 \leq Z \leq \text{MaxX}$) простым, достаточно проверить значение массива x в позиции Z . Если $x[Z]=1$, значит число Z – простое.

Задача 07_JanB. Qualified Primes [Kolstad/Ho, 2007]

Фермер Джон начал пометку коров последовательными простыми числами. Бесси заметила это и задумалась о вхождении различных цифр в эти метки. Помогите Бесси определить количество простых чисел в интервале (включая концы) $A..B$ ($1 \leq A \leq B \leq 4\,000\,000$; $B \leq A + 1\,000\,000$; один тест имеет $B \leq A + 2\,000\,000$), которые содержат некоторую данную цифру D . Простым называется положительное целое число, которое имеет ровно 2 делителя (1 и себя). Первые простые числа: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29.

Формат ввода:

* Строка 1: Три разделенных пробелом целых числа: A , B , D

Пример ввода (файл `qprime.in`):

10 15 3

Пояснения: сколько простых чисел в диапазоне 10..15 содержит цифру 3?

Формат вывода:

* Строка 1: Количество простых чисел в указанном диапазоне, которые содержат цифру D .

Пример вывода (файл `qprime.out`):

1

Пояснения: только число 13 в данном диапазоне содержит цифру '3'.

Для решения задачи вначале с помощью вышеописанной процедуры Resheto построим массив X, содержащий пометки, является ли число I простым ($x[i]=1$) или составным ($x[i]=0$). А затем переберем все числа в интервале от A до B и проверим каждое, является ли оно простым и содержит ли цифру D. Для последней проверки используется функция Yes(I,D), которая проверяет, содержится ли цифра D в числе I переводом числа в строку и вызовом встроенной функции pos.

```
{07_JanB. Qualified Primes}
const
  MaxX = 4000000;
var
  x : array [2..MaxX] of longint;
  i,num,a,b : longint;
  d : char;
Procedure Resheto;
var
  i,j : longint;
begin
  for i:=2 to MaxX do x[i]:=1;
  for i:=2 to trunc(sqrt(MaxX)) do
    if x[i]=1
      then begin
        j:=i;
        while (j+i)<=MaxX do
          begin
            j:=j+i;
            x[j]:=0;
          end;
        end;
  end;
function Yes(i:longint; d:char):boolean;
var
  s : string;
begin
  str(i,s);
  if pos(d,s)<>0
    then Yes:=true
    else Yes:=false;
end;
```

```

begin
  assign(input,'qprime.in'); reset(input);
  assign(output,'qprime.out'); rewrite(output);
  readln(a,b,d,d); Resheto;
  num:=0;
  for i:=a to b do
    if (x[i]=1) and Yes(i,d) then inc(num);
  writeln(num);
  close(input); close(output);
end.

```

Задача 05_OctB. Max Factor [Hal Burch, 2005]

Чтобы улучшить организацию своей фермы, фермер Джон пометил каждую из своих N ($1 \leq N \leq 5\,000$) коров уникальным номером в диапазоне от 1 до 20 000. К несчастью, он не предполагал, что коровам некоторые номера нравятся больше других. В частности, корова, чей номер имеет больший простой делитель, имеет больший социальный статус. Напомним, что простое число – это такое число, которое не имеет других делителей, кроме себя и 1. Например, число 7 – простое, а число 6 – не простое, поскольку имеет делители 2 и 3.

Вам дано N ($1 \leq N \leq 5\,000$) чисел в диапазоне от 1 до 20 000.

Определите такое из них, которое имеет наибольший простой делитель.

Формат ввода:

- * Строка 1: Одно целое число N
- * Строки 2.. $N+1$: номера, которые нужно проанализировать, по одному в строке

Пример ввода (файл bigfact.in):

```

4
36
38
40
42

```

Формат вывода:

- * Строка 1: Целое число с самым большим простым делителем. Если таких чисел несколько, выведите то из них, которое раньше появилось в исходном списке чисел.

Пример вывода (файл bigfact.out):

```

38

```

Пояснение к выводу: 19 – самый большой делитель числа 38. Никакое из других чисел не имеет простого делителя больше, чем 19.

И снова для решения задачи вначале построим массив пометок X, указывающих, является ли число l простым ($x[i]=1$) или составным ($x[i]=0$). Затем считываем числа, заданные по условию задачи, и для каждого из них с помощью процедуры MaxFactor ищем наибольший простой делитель соответствующего числа. По ходу запоминаем максимальный из найденных наибольших простых делителей и число, ему соответствующее, которое и выводим в качестве ответа по завершении ввода всех заданных чисел.

```
{05_OctB. Max Factor}
const
  MaxX = 20000;
var
  x : array [2..MaxX] of longint;
  i,n,max,a,num,mf : longint;

Procedure Resheto;
var
  i,j : longint;
begin
  for i:=2 to MaxX do x[i]:=1;
  for i:=2 to trunc(sqrt(MaxX)) do
    if x[i]=1
      then begin
        j:=i;
        while (j+i)<=MaxX do
          begin
            j:=j+i;
            x[j]:=0;
          end;
        end;
  end;

Procedure MaxFactor(a:longint; var mf:longint);
var
  j : longint;
begin
  j:=a div 2;
  while (j>1) and (((a mod j)<>0) or (x[j]=0)) do dec(j);
  if j>1
```

```

        then mf:=j
        else mf:=a;
end;

begin
    assign(input,'bigfact.in'); reset(input);
    assign(output,'bigfact.out'); rewrite(output);
    readln(N); max:=1; num:=1; Resheto;
    for i:=1 to N do
        begin
            readln(a); MaxFactor(a,mf);
            if mf>max then begin num:=a; max:=mf; end;
        end;
    writeln(num);
    close(input); close(output);
end.

```

Пояснение к процедуре MaxFactor(a,f), которая для заданного числа a находит его наибольший простой делитель mf.

```

Procedure MaxFactor(a:longint; var mf:longint);
var
    j : longint;
begin
    j:=a div 2;
    while (j>1) and (((a mod j)<>0) or (x[j]=0)) do dec(j);
    if j>1
        then mf:=j
        else mf:=a;
end;

```

В цикле от половины числа a (делителем не может быть число, которое больше половины a) проверяем все числа j до 2 включительно. Если a не делится на j или j не простое, берем число на 1 меньше.

Если встретилось такое число j, что a делится на j и j – простое ($x[j]=1$), то это j и есть наибольший простой делитель. Иначе – наибольший простой делитель есть само число a (других простых делителей не найдено).

Заключение

В данной статье приведен материал для обучения решению задач по информатике на тему «Решето Эратосфена».

Технической основой методики является разработанная инструментальная система дистанционного обучения (Distance Learning Belarus – <http://dl.gsu.by>). Все задачи, приведенные в статье, могут быть сданы в курсе «Методы алгоритмизации».

Литература

1. Долинский, М.С. Об опыте подготовки школьников Гомельской области к республиканским и международным олимпиадам по информатике / М.С. Долинский // Информатизация образования. – 2009. – № 1(54). – С.29-40.

2. Долинский, М.С. Система интернет-курсов дифференцированного обучения программированию школьников и студентов / М.С. Долинский, М.А. Кугейко // Информатизация образования. – 2010. – № 1(58). – С.58-68.

3. Долинский, М.С. Как учить думать школьников и студентов? / М.С. Долинский, М.А. Кугейко // Информатизация образования. – 2010. – № 2(59). – С.62-72.

4. Долинский, М.С. Технология развивающего дифференцированного обучения программированию младших школьников «с чистого листа» / М.С. Долинский, М.А. Кугейко // Информатизация образования. – 2010. – № 3(60). – С.12-20.

5. Долинский, М.С. Интернет-курс «Базовое программирование» как средство подготовки к областным олимпиадам по информатике / М.С. Долинский, М.А. Кугейко // Информатизация образования. – 2010. – № 4(61). – С.3-15.

6. Долинский, М.С. Развитие мышления младших школьников на основе флеш-заданий на рисование, раскраску и конструирование в системе DL.GSU.BY / М.С. Долинский, Ю.В. Решетько, М.А. Кугейко // Информатизация образования. – 2011. – № 1(62). – С.24-35.

7. Долинский, М.С. Какими должны быть задачи на олимпиадах по информатике / М.С. Долинский, М.А. Кугейко // Информатизация образования. – 2011. – № 1(62). – С.68-76.

8. Долинский, М.С. Флеш-шаблоны для создания заданий развивающего обучения / М.С. Долинский, Ю.В. Решетько, М.А. Долинская // Информатизация образования. – 2011. – № 2(63). – С.14-28.

9. Долинский, М.С. Конструирование интерактивных флеш-заданий на развитие мышления / М.С. Долинский, Ю.В. Решетько, М.А. Долинская // Информатизация образования. – 2011. – № 3(64). – С.21-33.

10. Долинский, М.С. Конструирование интерактивных флеш-заданий на развитие мышления на базе произвольных картинок / М.С. Долинский, Ю.В. Решетько, М.А. Долинская // Информатизация образования. – 2011. – № 4(65). – С.3-14.

11. Долинский, М.С. Конструирование интерактивных флеш-заданий на базе собственных танов / М.С. Долинский, Ю.В. Решетько, Н.С. Лебедько // Информатизация образования. – 2012. – № 1(66). – С.24-34.

12. Долинский, М.С. Конструктор интерактивных флеш-заданий как открытая система для создания электронных учебных пособий / М.С. Долинский, Ю.В. Решетько, М.А. Долинская, Н.С. Лебедько // Информатизация образования. – 2012. – № 2(67). – С.35-45.

13. Долинский, М.С. Электронное учебное пособие «Математика. Начальная школа» / М.С. Долинский, Ю.В. Решетько, Н.С.Лебедько // Информатизация образования. – 2012. – № 3(68). – С.30-42.

14. Долинский, М.С. Создание электронных учебных пособий для вузовских дисциплин с помощью конструктора флеш-заданий / М.С. Долинский, Ю.В. Решетько // Информатизация образования. – 2012. – № 4(69). – С.34-45.

15. Долинский, М.С. Интерактивная анимация в электронных учебных пособиях, создаваемых с помощью конструктора флеш-заданий / М.С. Долинский, Ю.В. Решетько, М.А. Долинская // Информатизация образования. – 2013. – № 1(70). – С.30-38.

16. Долинский, М.С. Учебный интернет-курс и перманентный интернет-конкурс «Математика 1-8 кл.» / М.С. Долинский, Ю.В. Решетько, М.А. Долинская // Информатизация образования. – 2013. – № 2(71). – С.38-47.

17. Долинский, М.С. Концептуальные основы и практика сквозного развивающего обучения информатике и программированию от детского сада до вуза / М.С. Долинский, Ю.В. Решетько, М.А. Долинская // Информатизация образования. – 2013. – № 3(72). – С.16-25.

18. Долинский, М.С. Об одном подходе к обучению программированию на первом курсе / М.С. Долинский, М.А. Долинская // Информатизация образования. – 2014. – № 1(73). – С.32-41

19. Долинский, М.С. Использование форума при обучении программированию первокурсников / М.С. Долинский // Информатизация образования. – 2014. – № 2(74). – С.22-34.

20. Долинский, М.С. Элементы теории чисел: системы счисления / М.С. Долинский // Информатизация образования. – 2015. – № 1(75). – С.14-28.

21. Долинский, М.С. Элементы теории чисел: битовая обработка / М.С. Долинский // Информатизация образования. – 2015. – № 2(76). – С.3-15.

Статья поступила 04.01.2016



РЕПОЗИТОРИЙ ГГУ ИМЕНИ Ф. СКОРИНЫ