

Зарубин М. М.
M. M. Zarubin

**ОБ ОДНОМ АЛГОРИТМЕ НЕСТАНДАРТНОЙ ТРАНСФОРМАЦИИ
ОДНОМЕРНОГО ЧИСЛОВОГО МАССИВА В ДВУМЕРНЫЙ МАССИВ**

**ON A NON-STANDARD TRANSFORMATION ALGORITHM OF ONE-MEASURED
NUMERICAL SET OF LATTICE POINTS TO A TWO-DIMENSIONAL SET OF LATTICE
POINTS**

Зарубин Михаил Михайлович – кандидат физико-математических наук, доцент кафедры прикладной математики и информатики Комсомольского-на-Амуре государственного технического университета (Россия, г. Комсомольск-на-Амуре). E-mail: zmm52@mail.ru.

Mr. Michael M. Zarubin – PhD in Physics and Mathematics, Associate Professor, Applied Mathematics and Informatics Department, Komsomolsk-on-Amur State Technical University (Russia, Komsomolsk-on-Amur). E-mail: zmm52@mail.ru.

Аннотация. Предложен алгоритм заполнения прямоугольной матрицы размеров $n \times m$ элементами сформированного одномерного числового массива по заданной непрерывной кривой. В отличие от традиционных алгоритмов решения задачи для некоторых модельных случаев (например, при заполнении прямоугольной матрицы по спирали) предложенный алгоритм не требует громоздких математических расчетов и легко адаптируется под конкретную геометрию нестандартного заполнения матрицы («кривой заполнения»). Это достигается организацией «препятствий» в процессе заполнения матрицы последовательными элементами исходного одномерного массива и переключением направления «движения» в соответствии с заданной схемой («кривой заполнения»). Приводится описание алгоритма для одной из модельных постановок задачи (заполнение по спирали по часовой стрелке, начиная с «северо-западного угла» матрицы) и его реализация на языке PASCAL (исходный текст программы и соответствующие скриншоты). В качестве исходного одномерного массива используется массив последовательных простых чисел, сгенерированных методом «решета Эратосфена».

Summary. An algorithm is proposed for filling a rectangular matrix of $n \times m$ elements of a formed one-dimensional numerical set of lattice points with a given continuous curve. Unlike traditional problem solving algorithms for some model cases (for example, when a rectangular matrix is spirally filled), the proposed algorithm does not require cumbersome mathematical calculations and is easily adapted to the specific geometry of non-standard matrix filling ("filling curve"). This is achieved by the organization of "obstacles" in the process of filling the matrix with consecutive incoming elements of the original one-dimensional set of lattice points and switching the direction of "motion", in accordance with the specified scheme ("filling curve"). The description of the algorithm for one of the model productions of the problem (filling in a clockwise spiral, starting from the "north-west corner" of the matrix) and its implementation in PASCAL (the source code of the program and the corresponding screenshots) is given. As the initial one-dimensional set of lattice points, set of lattice points of successive prime numbers generated by the "Eratosthenes" method is used.

Ключевые слова: алгоритм, матрица, нестандартное заполнение.

Key words: algorithm, matrix, non-standard filling.

УДК 510.51

Многие задачи прикладной математики требуют заполнения прямоугольной матрицы числами из некоторой совокупности.

Соответствующие числовые совокупности определяются конкретной задачей, а сами числа могут выражать самый разнообразный смысл: коэффициенты жесткости некоторой механической системы, физические параметры исследуемой модели в их численном выражении, коэффициенты системы обыкновенных или дифференциальных уравнений и т.д.

При этом под «стандартным заполнением» матрицы понимается ситуация, когда матрица заполняется построчно: по строкам – «слева направо», по столбцам – «сверху вниз».

Простая модификация «стандартного заполнения», когда меняются только направления заполнения строк или столбцов (например, когда заполнение идет «снизу вверх» или «справа налево», или «зигзагом»: по нечетным строкам – «слева направо», по четным – «справа налево») не представляет существенных трудностей и легко программируется.

В работе рассмотрен случай «существенно нестандартного» заполнения матрицы (по некоторой заданной непрерывной кривой) на примере ее заполнения по спирали. Предложенный при этом алгоритм может быть легко адаптирован и к другим «существенно нестандартным» случаям (изменению «кривой заполнения»).

Традиционный алгоритм заполнения матрицы по спирали (по часовой или против часовой стрелки) требует значительных математических усилий, так как по мере развертывания процесса заполнения матрицы меняется не только направление «движения», но и количество еще незаполненных элементов строки (столбца) (см., например, задачу 81.3 в [1]).

Предлагаемый в настоящей работе алгоритм позволяет избежать вышеописанных трудностей и легко запрограммировать соответствующее заполнение матрицы числами из заданного массива.

Отметим, что задача заполнения матрицы естественным образом разбивается на две подзадачи:

- 1) формирование требуемой числовой совокупности;
- 2) собственно «нестандартное заполнение» матрицы.

Подзадача 1 имеет самую широкую постановку, тесно связанную с изначально решаемой более общей задачей (для решения которой и формируется двумерный массив, то есть матрица), и выходит за пределы нашего рассмотрения. В качестве ее примера мы рассмотрим генерацию последовательных простых чисел методом «решета Эратосфена».

В данной работе нас будет интересовать подзадача 2, а именно трансформация сформированного одномерного массива в двумерный («нестандартное заполнение» матрицы).

В качестве примера конкретного решения подзадачи 2 – «нестандартного заполнения» матрицы мы рассмотрим задачу заполнения прямоугольной матрицы размеров $n \times m$ по спирали по часовой стрелке, начиная с «северо-западного угла» (как уже отмечалось выше, предложенный алгоритм легко адаптируется и к другим постановкам подзадачи 2).

Идея алгоритма заполнения состоит в следующем: встретив «препятствие», необходимо изменить направление «движения» (то есть заполнения элементов матрицы) в соответствии с поставленной подзадачей 2.

Например, для рассматриваемого нами случая – заполнения по спирали по часовой стрелке – требуется каждый раз поворачивать направо.

Закодировав направления движения следующим образом: 0 – «слева направо»; 1 – «сверху вниз»; 2 – «справа налево»; 3 – «снизу вверх», получим, что код направления следует увеличивать каждый раз на единицу, сбрасывая на ноль, по достижению максимального значения, равного четырем.

Главный момент реализации идеи алгоритма состоит в том, как организовать «препятствие».

Для этого обнуляем все элементы матрицы, окружив ее по периметру «забором» из единиц, то есть заводим вместо исходной матрицы размеров $n \times m$, матрицу размеров $(n + 2) \times (m + 2)$, заполняя нулями и единицами соответствующие элементы.

Заполнив «стартовый» элемент «малой матрицы» первым значением сформированной совокупности (в нашем случае, первым простым числом – 2), организуем «прощупывание» очередного элемента матрицы текущего направления (в нашем случае, направления с кодом 0 – «слева направо»).

Если «прощупываемый элемент» равен нулю, то заполняем данный элемент матрицы очередным значением сформированной числовой совокупности.

В противном случае («наткнулись на препятствие») изменяем соответствующим образом код направления и осуществляем «прощупывание» в новом направлении.

Во избежание заикливания (на последнем шаге заполнения матрицы «препятствие» возникнет в любом направлении, так как матрица будет уже полностью заполнена) мы контролируем текущее количество заполненных клеток матрицы с помощью счетчика (значение счетчика, равное $n \cdot m$, означает завершение процесса заполнения матрицы).

Отметим, что в случае, когда в исходной числовой совокупности могут содержаться элементы, равные 0 или 1, в качестве индикаторов не заполненных и, соответственно, заполненных элементов матрицы можно использовать любые не равные друг другу числа, заведомо не содержащиеся в исходном массиве.

Ниже приводится код программы на языке Pascal (откомпилированный в среде Borland Pascal 7.0), реализующей задачу заполнения по спирали прямоугольной матрицы размеров $n \times m$ последовательными простыми числами (сгенерированными методом «решета Эратосфена») по часовой стрелке, начиная с «северо-западного угла», и результат ее выполнения.

```
program SimplSpir;
uses CRT; {вспомогательный модуль для использования функций ClrScr – очистки экрана и Read-
Key – приостановки экрана пользователя}
const
  MaxCount=10000; {константа, регулирующая количество сгенерированных простых чисел}
  MaxDim=100; {константа, регулирующая максимальные размеры заполняемой матрицы}
label A; {метка для прерывания процесса заполнения матрицы при обнаружении недостаточного
количества сгенерированных простых чисел}
var
  N,M,i,j:byte; {переменные N, M – размеры заполняемой матрицы; i, j – рабочие переменные}
  Count,k:word; {переменная Count – количество сгенерированных простых чисел; k – порядковый
номер простого числа}
  MassivOfSimpleNumbers:array[1..MaxCount] of word; {массив для хранения сгенерированных
простых чисел}
  Matrix:array[0..MaxDim,0..MaxDim] of word; {матрица для заполнения}
  Direct:0..4; {код направления}
Procedure Eratosfen; {процедура генерации простых чисел методом "решета Эратосфена"}
var
  WorkMassiv:array[2..MaxCount] of boolean; {рабочая матрица для "просеивания" простых чисел
методом "решета Эратосфена"}
  i,j:word; {рабочие переменные}
begin
  for i:=2 to MaxCount do WorkMassiv[i]:=true;
  Count:=0;
  for i:=2 to MaxCount do
  begin
    if WorkMassiv[i] then
    begin
      Inc(Count);
      MassivOfSimpleNumbers[Count]:=i;
      j:=i;
      while j<= MaxCount do
      begin
        WorkMassiv[j]:=false;
        j:=j+i;
      end;
    end;
  end;
end;
```



```
end;
begin
  ClrScr; {очистка экрана}
  writeln('Введите размеры заполняемой матрицы:');
  write('Число строк N=');readln(N);
  write('Число столбцов M=');readln(M);
  Eratosfen; {генерация простых чисел методом "решета Эратосфена"}
  {Проверка достаточности количества сгенерированных простых чисел для заполнения матрицы
заданных размеров}
  if Count<N*M then
    begin
      writeln('Для заданных значений N и M сформировано недостаточное количество простых чи-
сел!');
      writeln('Требуется увеличить значение константы MaxCount в тексте программы!');
      ReadKey;
      goto A;
    end;
  Count:=N*M; {переменная Count уменьшается до значения, требуемого для заполнения матрицы
заданных размеров}
  {Подготовка матрицы к "нестандартному заполнению" – расстановка "препятствий" по периметру
матрицы и "расчистка" её рабочего пространства}
  for i:=1 to N do
    for j:=1 to M do Matrix[i,j]:=0;
  for j:=0 to M+1 do
    begin
      Matrix[0,j]:=1;
      Matrix[N+1,j]:=1;
    end;
  for i:=1 to N do
    begin
      Matrix[i,0]:=1;
      Matrix[i,M+1]:=1;
    end;
  {Инициализация переменных}
  Direct:=0;
  i:=1;
  j:=0;
  k:=0;
  {Заполнение матрицы}
  while k<Count do
    begin
      case Direct of
        0:if Matrix[i,j+1]=0 then
          begin Inc(j); Inc(k); Matrix[i,j]:=MassivOfSimpleNumbers[k] end
          else Inc(Direct);
        1:if Matrix[i+1,j]=0 then
          begin Inc(i); Inc(k); Matrix[i,j]:=MassivOfSimpleNumbers[k] end
          else Inc(Direct);
        2:if Matrix[i,j-1]=0 then
          begin Dec(j); Inc(k); Matrix[i,j]:=MassivOfSimpleNumbers[k] end
          else Inc(Direct);
```

```

3:if Matrix[i-1,j]=0 then
    begin Dec(i); Inc(k); Matrix[i,j]:=MassivOfSimpleNumbers[k] end
    else Direct:=0;

end;
end;
{Вывод результата}
writeln("Заполненная матрица имеет вид:");
for i:=1 to N do
    begin
        for j:=1 to M do write(Matrix[i,j]:5);
        writeln
    end;
ReadKey;
A:end.

```

```

Ярлык для Vr.bat
Введите размеры заполняемой матрицы:
Число строк N=20
Число столбцов M=15
Заполненная матрица имеет вид:
 2   3   5   7  11  13  17  19  23  29  31  37  41  43  47
317 331 337 347 349 353 359 367 373 379 383 389 397 401 53
313 683 691 701 709 719 727 733 739 743 751 757 761 409 59
311 677 1033 1039 1049 1051 1061 1063 1069 1087 1091 1093 769 419 61
307 673 1031 1321 1327 1361 1367 1373 1381 1399 1409 1097 773 421 67
293 661 1021 1319 1583 1597 1601 1607 1609 1613 1423 1103 787 431 71
283 659 1019 1307 1579 1783 1787 1789 1801 1619 1427 1109 797 433 73
281 653 1013 1303 1571 1777 1931 1933 1811 1621 1429 1117 809 439 79
277 647 1009 1301 1567 1759 1913 1949 1823 1627 1433 1123 811 443 83
271 643 997 1297 1559 1753 1907 1951 1831 1637 1439 1129 821 449 89
269 641 991 1291 1553 1747 1901 1973 1847 1657 1447 1151 823 457 97
263 631 983 1289 1549 1741 1889 1979 1861 1663 1451 1153 827 461 101
257 619 977 1283 1543 1733 1879 1987 1867 1667 1453 1163 829 463 103
251 617 971 1279 1531 1723 1877 1873 1871 1669 1459 1171 839 467 107
241 613 967 1277 1523 1721 1709 1699 1697 1693 1471 1181 853 479 109
239 607 953 1259 1511 1499 1493 1489 1487 1483 1481 1187 857 487 113
233 601 947 1249 1237 1231 1229 1223 1217 1213 1201 1193 859 491 127
229 599 941 937 929 919 911 907 887 883 881 877 863 499 131
227 593 587 577 571 569 563 557 547 541 523 521 509 503 137
223 211 199 197 193 191 181 179 173 167 163 157 151 149 139

```

```

Ярлык для Vr.bat
Введите размеры заполняемой матрицы:
Число строк N=50
Число столбцов M=50
Для заданных значений N и M сформировано недостаточное количество простых чисел!
Требуется увеличить значение константы MaxCount в тексте программы!

```

ЛИТЕРАТУРА

1. Брудно, А. Л. Олимпиады по программированию для школьников / А. Л. Брудно, Л. И. Каплан; под ред. Б. Н. Наумова. – М.: Наука. Главная редакция физико-математической литературы, 1985. – 96 с.
2. Немногин, С. А. Turbo Pascal / С. А. Немногин. – СПб.: Питер, 2003. – 492 с.