

ОЦЕНКА БЫСТРОДЕЙСТВИЯ АЛГОРИТМОВ ГЕНЕРАЦИИ ПЕРЕСТАНОВОК НА ОСНОВЕ ФАКТОРИАЛЬНЫХ ЧИСЕЛ

*А.Е. Горячев, аспирант,
Сумский государственный университет, г. Сумы*

Существуют различные алгоритмы генерации перестановок, использующие факториальные числа, обладающие рядом преимуществ. В статье ставится задача определения быстродействия этих алгоритмов, а также способ повышения быстродействия одного из них с целью определения наиболее эффективного.

Ключевые слова: *быстродействие, алгоритмы, генерация перестановок, факториальная система счисления.*

Існують різні алгоритми генерації перестановок, які використовують факторіальні числа, що мають низку переваг. У статті ставиться завдання визначення швидкодії цих алгоритмів, а також спосіб підвищення швидкодії одного з них з метою визначення найбільш ефективного.

Ключові слова: *швидкодія, алгоритми, генерація перестановок, факторіальна система числення.*

ПОСТАНОВКА ЗАДАЧИ

Перестановки широко используются на практике для решения различных задач, среди которых можно выделить задачи комбинаторной оптимизации, помехоустойчивой передачи данных и их защиты от несанкционированного доступа [1]. Существует множество методов получения перестановок. Одним из способов генерирования перестановок является использование факториальных чисел, близких к перестановкам по своей структуре и свойствам [2]. Методы, использующие для получения перестановок факториальные числа, обладают рядом преимуществ, таких как способность получения перестановок большого порядка, возможность генерирования как случайных перестановок, так и заданных их последовательностей.

В данной работе рассматриваются различные алгоритмы генерации перестановок, использующие факториальные числа. Ставится задача оценки их эффективности и выбора наиболее быстродействующего алгоритма.

РЕШЕНИЕ ЗАДАЧИ

Первый алгоритм генерации перестановок, использующих факториальные числа, основан на методе, описанном в [2]. Данный алгоритм выглядит следующим образом.

1. Запись факториального числа.
2. Цифра n -го разряда факториального числа без изменений считается первым элементом перестановки.
3. Проверка, была ли преобразована последняя цифра факториального числа. Если да, то переход к п.7. Если нет, то происходит переход к преобразованию следующей цифры факториального числа.
4. Цифра факториального числа сравнивается с наименьшим элементом перестановки из ранее полученных. Если она меньше данного элемента, то считается следующим элементом перестановки. Далее происходит переход к п.3.
5. Если цифра факториального числа больше элемента перестановки, либо равна ему, то она увеличивается на 1. Далее происходит проверка, производилось ли сравнение с последним из полученных ранее элементов

перестановки. Если нет, то происходит исключение из поиска минимального значения элемента перестановки, с которым производилось сравнение и осуществляется переход к п.4.

6. Если производилось сравнение с последним из полученных ранее элементов перестановки, то цифра факториального числа считается следующим элементом перестановки. Далее происходит переход к п.3.

7. Вывод полученной перестановки.

Быстродействие данного алгоритма определяет количество операций сравнений между преобразуемыми цифрами факториального числа и элементами перестановки. Определив данное число, можно судить об общей эффективности алгоритма.

Общее число операций сравнения элементов для рассматриваемого алгоритма будет определяться числом сравнений для преобразуемой в элемент перестановки цифры факториального числа с минимальным из ранее полученных элементов перестановки и числом сравнений между собой элементов перестановки с целью определения наименьшего из них. Для различных факториальных чисел число операций сравнения элементов будет различно, однако можно вычислить максимальную и минимальную величину и определить среднее её значение.

В том случае, когда производится наименьшее количество сравнений элементов перестановки, каждая цифра факториального числа, преобразуемого в перестановку, за исключением первой, сравнивается только с одним из ранее определённых элементов перестановки. Согласно алгоритму, если цифра факториального числа меньше минимального значения из ранее полученных элементов перестановки, она без изменений записывается как следующий элемент. Для нахождения минимального элемента необходимо сравнить между собой все полученные элементы перестановки. При этом второй элемент сравнивается только с первым, третий – с первым и вторым и так далее. Для n элементов при этом необходимо сделать $(n - 1) + (n - 2) + \dots + 2 + 1 = n \cdot (n - 1) / 2$ сравнений. Для перестановки порядка n требуется произвести сравнения от двух до $n-1$ элементов перестановки:

$$N_{\min} = (n - 1) \cdot (n - 2) / 2 + (n - 2) \cdot (n - 3) / 2 + \dots + 3 \cdot 2 / 2 + 2 \cdot 1 / 2. \quad (1)$$

Общее число сравнений элементов для рассматриваемого случая при порядке перестановки n будет определяться $n - 1$ сравнениями цифр факториального числа с наименьшим из ранее полученных элементов и N_{\min} сравнениями между собой элементов перестановки:

$$N_{o_min} = (n - 1) + N_{\min} = (n - 1) + (n - 1) \cdot (n - 2) / 2 + (n - 2) \cdot (n - 3) / 2 + \dots + 3 + 1 = n + (n - 1) \cdot (n - 2) / 2 + (n - 2) \cdot (n - 3) / 2 + \dots + 3. \quad (2)$$

Для случая, когда число операций сравнения максимально, цифра факториального числа, преобразуемого в перестановку, сравнивается со всеми ранее определёнными элементами перестановки в порядке возрастания их значений. Всего таких сравнений при порядке перестановки n будет $(n - 1) + (n - 2) + \dots + 2 + 1 = n \cdot (n - 1) / 2$.

При этом каждому такому сравнению предшествует нахождение минимального элемента перестановки из ранее определённых. Как было показано ранее, это требует $n \cdot (n - 1) / 2$ операций сравнения для n элементов. При каждом следующем нахождении минимального элемента не учитываются уже найденные в предыдущих циклах значения. Поэтому число сравнений будет уменьшаться на 1 до того момента, пока не будут выбраны все элементы. Общее число сравнений между собой ранее найденных элементов перестановок для n -й цифры факториального числа:

$$N_n = n \cdot (n - 1) / 2 + (n - 1) \cdot (n - 2) / 2 + \dots + 3 \cdot 2 / 2 + 2 \cdot 1 / 2. \quad (3)$$

Для всех цифр n -разрядного факториального числа количество сравнений между собой ранее найденных элементов перестановок будет следующим:

$$N_{\max} = \sum_{i=2}^n N_i = \left(\frac{2 \cdot 1}{2}\right) + \left(\frac{3 \cdot 1}{2} + \frac{2 \cdot 1}{2}\right) + \dots + \left(\frac{(n-1) \cdot (n-2)}{2} + \dots + \frac{3 \cdot 2}{2} + \frac{2 \cdot 1}{2}\right) + \left(\frac{n \cdot (n-1)}{2} + \dots + \frac{3 \cdot 2}{2} + \frac{2 \cdot 1}{2}\right). \quad (4)$$

Учитывая число операций сравнения цифр факториального числа с элементами перестановки, общее количество операций сравнения будет определяться следующим выражением:

$$\begin{aligned} N_{o_max} &= N_{\max} + \frac{n \cdot (n-1)}{2} = \left(\frac{2 \cdot 1}{2}\right) + \left(\frac{3 \cdot 1}{2} + \frac{2 \cdot 1}{2}\right) + \dots + \\ &+ \left(\frac{(n-1) \cdot (n-2)}{2} + \dots + \frac{3 \cdot 2}{2} + \frac{2 \cdot 1}{2}\right) + \left(\frac{n \cdot (n-1)}{2} + \dots + \right. \\ &+ \left. \frac{3 \cdot 2}{2} + \frac{2 \cdot 1}{2}\right) + \frac{n \cdot (n-1)}{2} = (n-1) + 3 \cdot (n-2) + \dots + \\ &+ 2 \cdot \frac{(n-1) \cdot (n-2)}{2} + \frac{n \cdot (n-1)}{2} + \frac{n \cdot (n-1)}{2} = \\ &= (n-1) + 3 \cdot (n-2) + \dots + 3 \cdot \frac{(n-2) \cdot (n-3)}{2} + \\ &+ (n-1) \cdot (n-2) + n \cdot (n-1). \end{aligned} \quad (5)$$

Пример 1. Определить среднее количество операций сравнения, требуемых для преобразования факториального числа в перестановку порядка $n = 16$ для первого метода.

Решение:

$$N_{o_min} = 16 + 15 \cdot 14 / 2 + 14 \cdot 13 / 2 + \dots + 6 + 3 = 575 ;$$

$$N_{o_max} = 15 + 14 \cdot 3 + 13 \cdot 6 + \dots + 2 \cdot 13 \cdot 12 + 3 \cdot 14 \cdot 13 / 2 + 15 \cdot 14 + 16 \cdot 15 = 3180 .$$

Среднее число операций сравнения:

$$N_{o_cp} = \frac{N_{o_max} + N_{o_min}}{2} = \frac{575 + 3180}{2} = 1877,5 .$$

Другой известный алгоритм получения перестановок на основе факториальных чисел, описанный в [1], выглядит следующим образом.

1. Запись факториального числа.
2. Выбирается вторая цифра факториального числа.
3. Выбранная цифра факториального числа сравнивается со всеми предыдущими. Те цифры, которые больше выбранной, увеличиваются на 1, остальные остаются без изменений.
4. Выбранная цифра факториального числа увеличивается на 1.
5. Проверка, была ли выбрана последняя цифра факториального числа. Если да, то происходит переход к п.6. Если нет, то выбирается следующая по возрастанию номера цифра факториального числа. Далее происходит переход к п.3.

6. Вывод полученной перестановки.

Для данного метода получения перестановок число сравнений, необходимых для преобразования каждого факториального числа в перестановку, постоянно. Данное число будет определяться только порядком перестановки. Общее количество сравнений, необходимых для получения перестановки при использовании этого метода будет определяться следующим образом: для цифры младшего разряда преобразуемого факториального числа число сравнений равно нулю, следующая цифра сравнивается только с младшей и так далее. Цифра в старшем разряде факториального числа сравнивается со всеми предыдущими. Таким образом, общее число сравнений для перестановки порядка n будет равно:

$$N_o = (n - 1) + (n - 2) + \dots + 2 + 1 = n \cdot (n - 1) / 2. \quad (6)$$

Пример 2. Определить количество операций сравнения, требуемых для преобразования факториального числа в перестановку порядка $n = 16$ для второго метода.

Решение:

$$N_o = 15 + 14 + \dots + 2 + 1 = 16 \cdot 15 / 2 = 120.$$

Данная цифра значительно ниже, чем среднее и минимальное число сравнений для первого рассмотренного метода. Недостатком второго метода по сравнению с первым является только невозможность передачи сформированных элементов перестановки до окончания преобразования, однако данный недостаток незначителен по сравнению с выигрышем в быстродействии более чем на порядок.

Основная потеря быстродействия первого метода наблюдается при определении минимальных значений преобразованных элементов перестановки (слагаемое N_{\min} в формуле (1) и N_{\max} в формуле (4)), следовательно, значительно повысить быстродействие этого метода можно с помощью ввода процесса сортировки элементов перестановки параллельно процессу преобразования. Этот процесс может осуществляться следующим образом.

Для сортировки элементов перестановки выделяется область памяти, состоящая из $n-1$ ячеек, в каждой из которых может храниться 1 из элементов. Вначале процесса преобразования факториального числа в перестановку цифра старшего его разряда, соответствующая первому элементу перестановки, заносится в первую ячейку памяти. Следующая цифра факториального числа сравнивается с элементом перестановки, содержащимся в первой ячейке памяти. Если она больше данного элемента, либо равна ему, то увеличивается на единицу и записывается во вторую ячейку памяти. Если же цифра факториального числа меньше элемента перестановки, то она записывается без изменений в первую ячейку памяти, элемент перестановки, хранящийся в первой ячейке, при этом записывается во вторую. Далее происходит переход к преобразованию следующего элемента.

В общем случае процесс преобразования k -й цифры факториального числа будет происходить следующим образом. Эта цифра сравнивается последовательно со всеми элементами перестановки, хранящимися в памяти, начиная с первой ячейки. Если цифра факториального числа больше либо равна элементу перестановки, то она увеличивается на 1 и сравнивается с элементом перестановки из следующей ячейки памяти. Если же цифра факториального числа меньше элемента перестановки, то она без изменений записывается в ту ячейку памяти, в которой был записан элемент перестановки, с которым происходило сравнение. При этом происходит перезапись значений элементов перестановки в

следующую ячейку памяти, начиная со сравниваемого элемента. В случае, когда цифра факториального числа больше либо равна последнему ($k-1$)-му элементу перестановки из ранее полученных, она увеличивается на 1 и записывается в первую незаполненную ячейку памяти (k -ю для k -го элемента).

Таким образом, для k -й цифры факториального числа требуется совершить 1 операцию сравнения в минимальном случае, когда она меньше элемента перестановки из первой ячейки памяти, и $k-1$ операций сравнения в максимальном случае, когда она больше элемента перестановки из ($k-1$)-й ячейки памяти. Общее количество операций сравнения при преобразовании факториального числа в перестановку порядка n с применением сортировки элементов будет следующим.

Для случая минимального количества операций сравнения

$$N_{o_min} = n - 1. \quad (7)$$

Для случая максимального количества операций сравнения

$$N_{o_max} = (n - 1) + (n - 2) + \dots + 2 + 1 = n \cdot (n - 1) / 2. \quad (8)$$

Пример 3. Определить количество операций сравнения, требуемых для преобразования факториального числа в перестановку порядка $n = 16$ методом с использованием сортировки элементов перестановки.

Решение:

$$N_{o_min} = 15;$$

$$N_{o_max} = 15 + 14 + \dots + 2 + 1 = 120;$$

$$N_{o_cp} = \frac{N_{o_max} + N_{o_min}}{2} = \frac{15 + 120}{2} = 67,5.$$

Из формул (8) и (9) видно, что первый метод с применением сортировки элементов перестановки в максимальном случае показывает быстроедействие, аналогичное второму методу, а в минимальном случае значительно его превосходит. В среднем, быстроедействие первого метода также будет выше быстроедействия второго метода. Учитывая то, что первый метод также даёт возможность передачи сформированных элементов перестановки до окончания преобразования, его применение является более эффективным.

Алгоритм генерации перестановок на основе факториальных чисел с использованием сортировки элементов перестановки.

1. Запись факториального числа. Установка в нулевое значение ячеек $M_1 - M_{n-1}$ области памяти, предназначенной для сортировки элементов перестановки.

2. Цифра n -го разряда факториального числа без изменений записывается в ячейку памяти M_1 и считается первым элементом перестановки.

3. Проверка, была ли преобразована последняя цифра факториального числа. Если да, то переход к п.7. Если нет, то происходит переход к преобразованию следующей цифры факториального числа.

4. Цифра факториального числа сравнивается с элементом перестановки, хранящимся в ячейке памяти M_i ($i = 1, 2, \dots, n-1$). Если она меньше данного элемента, то считается следующим элементом перестановки и записывается в ячейку M_i , величина, хранящаяся в M_i , записывается при этом в ячейку M_{i+1} , из M_{i+1} в M_{i+2} и так далее до

заполнения первой из ещё незаполненных ячеек памяти. Далее происходит переход к п.3.

5. Если цифра факториального числа больше элемента перестановки, либо равна ему, то она увеличивается на 1. Далее происходит проверка, производилось ли сравнение с последним из полученных ранее элементов перестановки. Если нет, то происходит увеличение номера i ячейки памяти, в которой содержится элемент перестановки, с которым сравнивается цифра факториального числа, на 1 и осуществляется переход к п.4.

6. Если производилось сравнение с последним из полученных ранее элементов перестановки, то цифра факториального числа считается элементом перестановки и записывается в первую из незаполненных ячеек памяти. Происходит переход к п.3.

7. Вывод полученной перестановки.

ВЫВОДЫ

Проведённая в работе оценка быстродействия алгоритмов генерации перестановок на основе факториальных чисел позволяет сделать вывод, что второй метод является гораздо более быстродействующим, чем первый, однако существует возможность значительно повысить быстродействие первого метода. С применением сортировки элементов перестановки быстродействие первого метода в среднем превышает быстродействие второго, кроме того, данный метод позволяет осуществлять вывод сформированных элементов перестановки в процессе преобразования, что делает его более эффективным.

SUMMARY

SPEED ESTIMATION OF PERMUTATION GENERATION ALGORITHMS BASED ON FACTORIAL NUMBERS

*A.E. Goryachev,
Sumy State University, Sumy*

The article deals with various algorithms for permutation generation using factorial numbers. The effectiveness of these algorithms with purpose to determine the fastest of them is under investigation.

Key words: *performance, algorithms, generation of permutations, factorial counting system.*

СПИСОК ЛИТЕРАТУРЫ

1. Рейнгольд Э. Комбинаторные алгоритмы: теория и практика / Э. Рейнгольд, Ю. Нивергельт, Н. Део. – М.: Изд-во "Мир", 1980. – 477 с.
2. Борисенко А.А. Электронная система генерации перестановок на базе факториальных чисел / А.А. Борисенко, И.А. Кулик, А.Е. Горячев // Вісник СумДУ. Серія Технічні науки. – 2007. – №1. – С.183 – 188.

Поступила в редакцию 14 апреля 2010 г.