УДК 681.3.06

МЕТОД ПЕРЕБОРА ПЕРЕСТАНОВОК НА ОСНОВЕ ФАКТОРИАЛЬНЫХ ЧИСЕЛ

А.Е Горячев,
Сумский государственный университет

Для решения задачи полного перебора перестановок определённой длины может использоваться алгоритм генерации перестановок на основе факториальных чисел. В статье ставится задача повышения быстродействия этого алгоритма применительно к задаче перебора перестановок за счёт копирования одинаковых элементов подряд идущих перестановок.
Ключевые слова: *генерация перестановок, перебор перестановок, алгоритмы, факториальные числа*

Для вирішення завдання повного перебору перестановок певної довжини може використовуватися алгоритм генерації перестановок на основі факторіальних чисел. У статті ставиться завдання підвищення швидкодії цього алгоритму щодо задачі перебору перестановок за рахунок копіювання однакових елементів перестановок, що йдуть поспіль.
Ключові слова: *генерація перестановок, перебір перестановок, алгоритми, факторіальні числа*

ПОСТАНОВКА ЗАДАЧИ

Такой комбинаторный объект как перестановки находит широкое применение при решении задач комбинаторной оптимизации, помехоустойчивой передачи данных и их защиты от несанкционированного доступа [1]. Часто на практике актуальным является полный либо частичный перебор перестановок с числом элементов n в порядке увеличения их номера (лексикографический порядок). К примеру, при решении различных задач комбинаторной оптимизации методом грубой силы требуется получить все возможные решения и выбрать из них наилучшее, для чего необходимо перечислить все возможные перестановки определённой длины [2].

Существует несколько методов получения перестановок. Одним из них является метод генерации перестановок на основе факториальных чисел [3]. Преимуществом использования факториальных чисел для перебора перестановок является возможность быстрого перебора факториальных чисел в порядке возрастания их значений [4], последовательное получение элементов перестановок в процессе преобразования, получение перестановок большой длины, выбор начального и конечного значений номера генерируемых перестановок с помощью ограничений значений факториальных чисел.

Алгоритм преобразования факториальных чисел в перестановки, использующий сортировку элементов перестановки, был рассмотрен в [5]. В данной работе ставится задача повышения быстродействие этого алгоритма генерации перестановок применительно к задаче их перебора.

РЕШЕНИЕ ЗАДАЧИ

Возможность повышения быстродействия алгоритма, генерирующего перестановки на основе факториальных чисел, при переборе перестановок в лексикографическом порядке обеспечивается следующими его особенностями:

1. Процесс преобразования факториального числа в перестановку ведётся поразрядно, начиная со старшего разряда факториального числа и заканчивая младшим [3].

2. При переборе факториальных чисел, их значения увеличиваются каждый раз на единицу, причём прибавление единицы идёт в первом разряде. Старшие разряды при этом, в случае отсутствия переносов, остаются неизменными [4].

Исходя из данных особенностей можно сделать вывод, что для подряд идущих факториальных чисел несколько первых элементов соответствующих им перестановок могут совпадать. Если совпадают значения старших разрядов данных чисел, преобразование данных значений в элементы перестановки будет одинаковым и, соответственно, на выходе будут получаться одни и те же значения элементов перестановки. Следовательно, для сокращения времени, требующегося для генерирования подряд идущих перестановок, можно часть элементов следующей перестановки не получать путём преобразования из цифр разрядов факториального числа, а копировать из предыдущей сгенерированной перестановки.

Пример: В идущих подряд по возрастанию значений факториальных числах 3 0 1 1 1 0 и 3 0 1 2 0 0, которым соответствуют десятичные номера 369 и 370, одинаковы значения трёх старших разрядов: 3, 0, 1. Данным числам соответствуют перестановки 3 0 2 4 5 1 и 3 0 2 5 1 4, в которых значения трёх первых элементов 3, 0, 2 также одинаковы. Поэтому, после преобразования факториального числа 3 0 1 1 1 0 в перестановку 3 0 2 4 5 1, для получения следующей по номеру перестановки, потребуется преобразовать только 3 младших разряда 2 0 0 факториального числа 3 0 1 2 0 0 в элементы 5 1 4, а три первых элемента 3 0 2 взять из предыдущей перестановки.

Для применения данного метода необходимо выполнение следующих условий:

1. Известны значения элементов предыдущей по номеру перестановки.
2. Известно число старших разрядов факториального числа, которые при увеличении предыдущего на единицу остались неизменными.
3. В процессе преобразования факториального числа в перестановку, также осуществляется сортировка сгенерированных элементов. Следовательно, в случае переноса части элементов из предыдущей по номеру перестановки в следующую, требуется также перенести значения ячеек памяти, в которых хранятся значения тех же элементов в порядке возрастания их значений либо заново произвести сортировку перенесённых элементов (что приведёт к снижению быстродействия процесса преобразования).

Таким образом, при преобразовании n-разрядного факториального числа в перестановку длины n требуется заранее генерировать следующее по счёту факториальное число для определения числа m одинаковых разрядов. Также необходимо наличие двух дополнительных областей памяти, каждая из которых содержит m ячеек. В первую область будут записываться m значений элементов перестановки, общие для рассматриваемых факториальных чисел. Во вторую область необходимо записать те же значения, но из ячеек памяти сортировки в порядке возрастания их значений.

С учётом описанных выше условий алгоритм перебора перестановок на основе факториальных чисел будет выглядеть следующим образом:

1. Получение первого текущего n-разрядного факториального числа F_0. Установка в нулевое значение ячеек M_{p1} – $M_{p(n-2)}$ области памяти, предназначенной для хранения повторяющихся элементов перестановки и ячеек M_{s1} – $M_{s(n-2)}$ области памяти, предназначенной для хранения повторяющихся элементов, отсортированных в порядке возрастания значений. Установка значений параметров $m_0 = 0$, $m_1 = 0$, обозначающих

число старших разрядов, значения которых одинаковы в идущих подряд факториальных числах.

2. Получение следующего факториального числа F_1 путём увеличения на единицу текущего.

3. Определение m_1 старших разрядов, значения в которых одинаковы для обоих факториальных чисел.

4. Запись m_0 значений из ячеек памяти M_{p1} – M_{pm0} как элементов перестановки P_n – $P_{n\text{-}m0}$, из ячеек M_{s1} – M_{sm0} в ячейки памяти сортировки M_1 – M_{m0}. Если $m_0 < m_1$, то преобразование текущего факториального числа F_0 в перестановку P длины n в соответствии с алгоритмом генерации перестановок с использованием сортировки элементов перестановки, начиная с n-m_0 разряда до n-m_1 разряда. Если нет, то переход к п.7.

5. Запись значений преобразованных элементов перестановки $P_{n\text{-}m0\text{-}1}$ – $P_{n\text{-}m1}$ в ячейки памяти M_{pm0+1} – M_{pm1}, запись значений ячеек памяти сортировки M_{m0+1} – M_{m1} в ячейки M_{sm0+1} – M_{sm1}.

6. Установка значения $m_0 = m_1$.

7. Преобразование оставшихся разрядов числа F_0 в перестановку P.

8. Вывод перестановки P.

9. Проверка, была ли получена последняя перестановка из генерируемой последовательности. Если да, то происходит оповещение об окончании процесса перебора. Если нет, то значение следующего факториального числа F_1 становится текущим F_0.

10. Если значение F_0 соответствует последнему номеру перестановки, которую необходимо сгенерировать, то параметру m_1 присваивается значение 0 и происходит переход к п.4. Если нет – переход к п.2.

В случае, когда необходимо осуществить полный перебор всех $n!$ перестановок, первым факториальным числом будет число вида F_n = 0 0 0 ... 0 0 0. Данному числу соответствует перестановка P_n = 0 1 2 ... n–3 n–2 n–1, значения элементов которой не нужно сортировать, так как они расположены в порядке возрастания значений. Эти значения могут быть заранее записаны в память.

В рассматриваемом алгоритме при переносе из предыдущей перестановки в следующий используются только m одинаковых элементов, однако возможно также осуществлять перенос (m+1)-го элемента, с некоторыми ограничениями. В идущих подряд факториальных числах после m одинаковых значений в старших разрядах, в (m+1)-м разряде значения различаются на 1 либо в результате переноса из младших разрядов, либо в результате добавления 1 к первому разряду. Если значение (m+1)-й цифры разряда факториального числа $F_{0(m+1)}$ перешло в результате k сравнений в элемент перестановки $P_{0(m+1)}$, то для следующего факториального числа значение (m+1)-й цифры разряда $F_{1(m+1)} = F_{0(m+1)} + 1$ в результате k сравнений перейдёт в элемент перестановки $P_{1(m+1)} = P_{0(m+1)} + 1$. Далее, необходимо произвести дополнительные сравнения элемента $P_{1(m+1)}$ с ранее полученными элементами перестановки, начиная с k+1 ввиду того, что значение $P_{1(m+1)}$ было увеличено на 1.

Данный приём позволяет сократить число сравнений при преобразовании факториального числа в перестановку на k, однако требует выполнения дополнительных арифметических операций, что усложняет алгоритм перебора.

Повышение быстродействия преобразования факториальных чисел в перестановки при использовании вышеописанного алгоритма перебора перестановок будет зависеть от числа m разрядов, одинаковых для идущих подряд факториальных чисел. Для разных пар чисел данная величина будет отличаться, следовательно, для определения прироста быстродействия необходимо определить и просуммировать параметр m для каждой пары из диапазона $n!$ факториальных чисел разрядностью n. Для оценки среднего значения выигрыша в быстродействии данную сумму нужно разделить на $n!$.

Добавление 1 при переборе факториальных чисел происходит в первом разряде (нулевой разряд всегда содержит 0). Следовательно, при отсутствии переносов в старшие разряды, максимальное значение параметра m для n-разрядного факториального числа $m = n - 2$. В случае, когда в результате переносов прибавляется 1 к цифре старшего разряда, $m = 0$.

Частота переносов из младшего разряда в старший определяется весом данного разряда. Для факториального числа вес j-того разряда, согласно нумерационной функции, равняется $j!$, т.е. для первого разряда – 1, для второго – 2, для третьего – 6, и так далее. Следовательно, перенос из первого разряда во второй будет происходить каждый второй такт перебора факториальных чисел, из второго в третий – каждый шестой и так далее. Перенос в старший $(n-1)$-й разряд будет происходить каждый $(n-1)!$-й такт. Учитывая, что общее число n-разрядных факториальных чисел равняется n!, число переходов, при которых будет происходить перенос в старший разряд, будет равняться: $T_{n-1} = n!/(n-1)! = n$.

Аналогично для числа переходов, при которых будет происходить перенос в $(n-2)$-й разряд, можно записать $T_{n-2} = n!/(n-2)! = n\cdot(n-1)$. Однако необходимо учитывать, что в данное число также входят переходы с переносом в старший $(n-1)$-й разряд, когда происходит переполнение $(n-2)$-го разряда, поэтому необходимо из величины $n\cdot(n-1)$ вычесть T_{n-1}:

$$T_{n-2} = n!/(n-2)! - n!/(n-1)! = n\cdot(n-1) - n = n\cdot(n-2).$$

Подобное выражение можно записать для любого разряда факториального числа: число переходов, при которых будет происходить перенос в j-й разряд, будет равняться общему числу факториальных чисел заданной разрядности, делённому на вес j-го разряда, за вычетом числа переходов, при которых будет происходить перенос в (j+1)-й разряд. Данное выражение также справедливо для первого разряда, переносом в него будет считаться добавление 1.

$$T_j = \frac{n!}{j!} - \frac{n!}{(j+1)!} = n!\left(\frac{j+1}{(j+1)!} - \frac{1}{(j+1)!}\right) = n!\cdot\frac{j}{(j+1)!}. \quad (1)$$

Далее необходимо определить для каждого T_j количество сравнений, необходимых для преобразования факториального числа в перестановку после переноса из предыдущей перестановки первых n–1–j элементов. Данную величину можно получить исходя из следующих соображений: количество сравнений N, необходимое для преобразования старших разрядов факториального числа в элементы перестановки ограничено и не зависит от разрядности числа n. Для старшего (n–1)-го разряда $N_{(n-1)} = 0$, для (n–2)-го $N_{(n-2)} = 0 \dots 1$, для (n–3)-го $N_{(n-3)} = 0 \dots 2$ и так далее. Следовательно, процесс преобразования первых n–1–j цифр факториального числа можно рассматривать как преобразование в перестановку факториального числа разрядности n–j–1. Среднее количество сравнений для данного преобразования рассматривалось в [5]. Эта величина вычитается из среднего числа сравнений для преобразования n-разрядного факториального числа, которое также было найдено в [5]. Сумма произведений величины из формулы (1) на данную разность, произведённая по всем возможным значениям величины j, даст приближённое значение общего числа сравнений, необходимых для получения всех n! перестановок при полном переборе.

$$N_{о} = \sum_{j=1}^{n-1}(T \cdot (N_{о}\ (n) - N\ (n-j-1))) \tag{2}$$

Подставив в формулу (2) значения T_j, из формулы (1) и $N_{о\ ср}(n)$, $N_{о\ ср}(n–j–1)$ из формулы, записанной в [5] получим:

$$N_{о} = \sum_{j=1}^{n-1}(n! \cdot \frac{j}{(j+1)!} \cdot (\frac{(n-1)\cdot(n+2)}{4} - \frac{(n-j-1-1)\cdot(n-j-1+2)}{4})) =$$

$$= n! \cdot \sum_{j=1}^{n-1}(\frac{j}{(j+1)!} \cdot (\frac{(n-1)\cdot(n+2)}{4} - \frac{(n-j-2)\cdot(n-j+1)}{4})) =$$

$$= n! \cdot \sum_{j=1}^{n-1}(\frac{j}{(j+1)!} \cdot \frac{n^2-n+2n-2-(n^2-j\cdot n+n-j\cdot n+j^2-j-2n+2j-2)}{4}) =$$

$$= n! \cdot \sum_{j=1}^{n-1}(\frac{j}{(j+1)!} \cdot \frac{2n+2nj-j^2-j}{4}) = n! \cdot \sum_{j=1}^{n-1}(\frac{j}{(j+1)!} \cdot \frac{(j+1)(2n-j)}{4}) = n! \cdot \sum_{j=1}^{n-1}\frac{(2n-j)}{4\cdot(j-1)!}.$$

$$N_{о} = n! \cdot \sum_{j=1}^{n-1}\frac{(2n-j)}{4\cdot(j-1)!}. \tag{3}$$

Для определения усреднённого числа операций сравнения, производимых при переборе перестановок длины n, необходимо значение общего числа сравнений, совершаемых при переборе, из формулы (3) разделить на количество генерируемых перестановок, то есть на n!.

$$N_{о} = \frac{N_{о}}{n!} = \sum_{j=1}^{n-1}\frac{(2n-j)}{4\cdot(j-1)!}. \tag{4}$$

С помощью величины $N_{оп\ ср}$ (4) можно сравнить влияние длины перестановок на количество операций сравнения для метода перебора перестановок и метода генерации перестановок с использованием сортировки элементов перестановки, описанного в [5]. Результаты данного исследования показаны на рисунке 1, где $N_{о\ ср}$ – количество операций сравнения для алгоритма генерации перестановок с использованием сортировки элементов, $N_{оп\ ср}$ – количество операций сравнения для алгоритма перебора перестановок.

Как следует из данных рисунка 1, среднее значение числа сравнений для алгоритма перебора перестановок линейно зависит от их длины, в отличие от алгоритма генерации перестановок, для которого данная зависимость носит полиномиальный характер. Следовательно, разница в среднем количестве операций сравнения для рассматриваемых алгоритмов значительно возрастает с ростом числа элементов перестановок, что позволит достичь значительного выигрыша в быстродействии.

Следует отметить, что величины, описанные в формулах (2) и (3) носят приближённый характер и могут отличаться от реальных величин в особенности при малых значениях длины перестановок. Было проведено моделирование алгоритмов генерации и перебора перестановок для более точной оценки и сравнения их быстродействия. Результаты исследования (рисунок 2) показали неточность значений, полученных ранее с помощью формулы (4), ввиду использования приближённого значения $N_{o\ cp}$. На рисунке 2 $N_{ом\ cp}$, $N_{опм\ cp}$ – количество операций сравнения алгоритмов генерации и перебора перестановок соответственно, полученные с помощью моделирования алгоритмов.

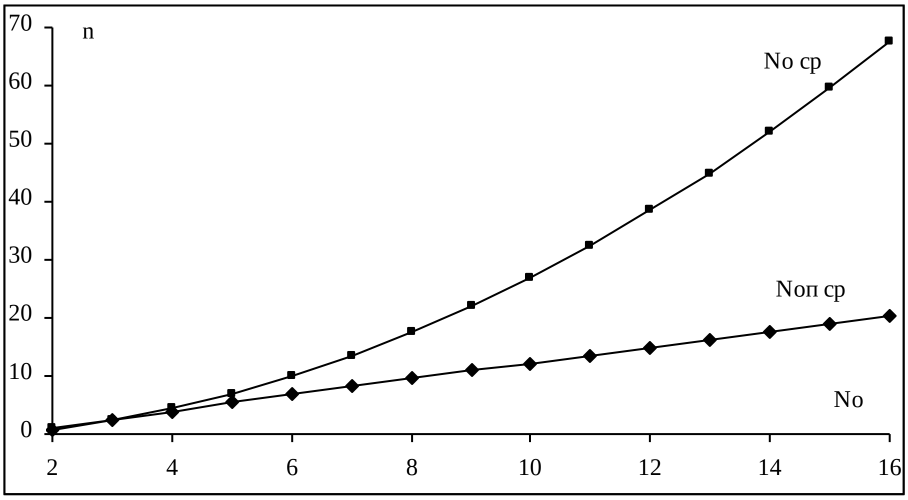

Рисунок 1 – Сравнение зависимости числа операций сравнения от длины перестановок для методов перебора и генерации перестановок

Тем не менее, зависимость значения, полученного моделированием от длины перестановок, также как и полученного по формуле (4), носит линейный характер. По значению $N_{опм\ cp}$ также значительно ниже, чем значение среднего числа операций сравнения для метода генерации перестановок $N_{ом\ cp}$.

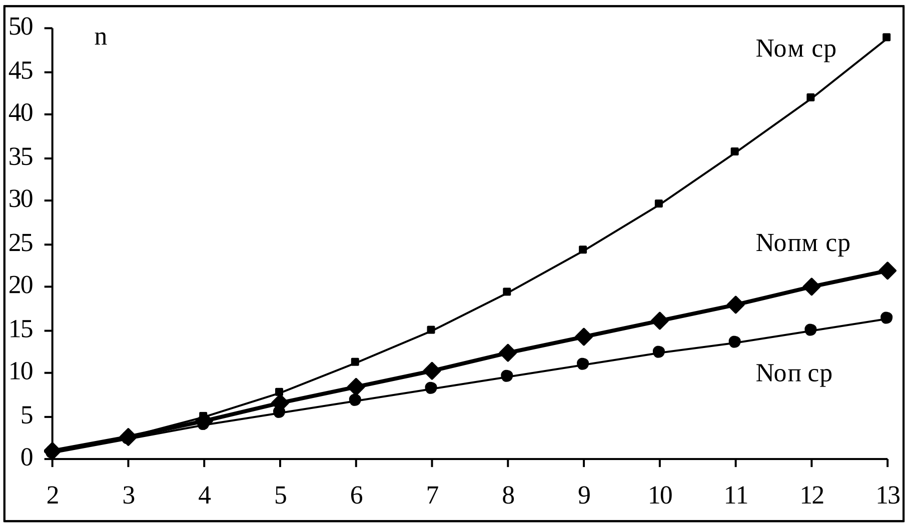

Рисунок 2 – График зависимости значений среднего числа операций сравнения методов генерации и перебора перестановок от длины перестановок

ВЫВОДЫ

Предложенный способ повышения быстродействия алгоритма, генерирующего перестановки на основе факториальных чисел, для решения задачи перебора всех перестановок позволяет значительно сократить количество операций, необходимых для преобразования факториального числа в перестановку. Получена линейная зависимость количества операций сравнения элементов от длины перестановки, что означает повышение выигрыша в быстродействии с ростом количества элементов перестановок.

SUMMARY

PERMUTATION ENUMERATION METHOD BASED ON FACTORIAL NUMBERS

A.E. Goryachev,
Sumy State University

To solve the problem of complete enumeration of permutations of a certain length algorithm for generating permutations based on the factorial numbers can be used. The paper seeks to improve performance of this algorithm for the enumeration of permutations task by copying the same elements of consecutive permutations.

Keywords: permutations generation, permutation enumeration, algorithms, factorial numbers

СПИСОК ЛИТЕРАТУРЫ

1. Рейнгольд Э. Комбинаторные алгоритмы: теория и практика / Э. Рейнгольд, Ю. Нивергельт, Н. Део. – М.: Мир, 1980. – 477 с.
2. Левитин А.В. Алгоритмы: введение в разработку и анализ/ А.В. Левитин. – М.: Вильямс, 2006. - 576 с.
3. 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.
4. Горячев А.Е. Метод получения факториальных чисел // Научно-техническая конференция преподавателей, сотрудников, аспирантов и студентов факультета ЭлИТ СумГУ, 2009.
5. Горячев А.Е. Оценка быстродействия алгоритмов генерации перестановок на основе факториальных чисел // Вісник СумДУ. Серія «Технічні науки». – 2010. – №1. – С. 62 – 67.

Поступила в редакцию 22 ноября 2010 г.