УДК 519.85

И.В. Гребенник, А.С. Литвиненко

ХНУРЭ, г. Харьков, Украина, igor.grebennik@nure.ua;
ХНУРЭ, г. Харьков, Украина, litvinenko1706@gmail.com

ГЕНЕРАЦИЯ ПЕРЕСТАНОВОК С ЧАСТИЧНО ОПРЕДЕЛЕННЫМ ПОРЯДКОМ СЛЕДОВАНИЯ ЭЛЕМЕНТОВ

В статье рассматривается задача генерации перестановок, для которых порядок следования (отношения «больше» или «меньше») порождающих элементов в каждой перестановке задан частично, то есть только для некоторых из них. Описан алгоритм, позволяющий сгенерировать все перестановки, удовлетворяющие частично заданному порядку следования. Описанный алгоритм является универсальным и может быть использован для генерации других комбинаторных конфигураций.

КОМБИНАТОРИКА, ПЕРЕСТАНОВКА, ГЕНЕРАЦИЯ, ПОРЯДОК СЛЕДОВАНИЯ ЭЛЕМЕНТОВ, МНОЖЕСТВО СПУСКА

Гребеннік І.В., Литвиненко О.С. Генерація перестановок з частково заданим порядком слідування елементів. В статті розглядається задача генерації перестановок, для яких порядок слідування (відношення «більше» або «менше») утворюючих елементів у кожній перестановці задано частково, тобто тільки для деяких з них. Запропоновано алгоритм, що дозволяє згенерувати всі перестановки, що задовольняють частково заданому порядку слідування. Описаний алгоритм є універсальним і може бути застосований для генерації інших комбінаторних конфігурацій.

КОМБІНАТОРИКА, ПЕРЕСТАНОВКА, ГЕНЕРАЦІЯ, ПОРЯДОК СЛІДУВАННЯ ЕЛЕМЕНТІВ, МНОЖИНА СПУСКУ

Grebennik I.V., Lytvynenko O.S. Generating permutations with partially defined order of elements. Enumeration and generation of permutations where order of elements is defined partially is considered. Algorithm for generating all such permutations is introduced. Described algorithm is quite universal and can be used for generating other combinatorial sets.

COMBINATORICS, PERMUTATION, GENERATION, ORDER OF ELEMENTS, SET OF DESCENT

Введение

Генерация комбинаторных множеств является направлением многих научных и прикладных исследований, например [1-3]. При этом основное внимание уделяется генерации классических комбинаторных множеств: перестановок, сочетаний, размещений и т.д. Однако, для решения некоторых научных и прикладных задач требуется решить задачу генерации неклассических комбинаторных множеств либо новых с точки зрения их комбинаторной структуры, либо обладающих дополнительными свойствами по сравнению с классическими (например, [4-5]).

Одним из таких неклассических комбинаторных множеств являются перестановки с заданными спусками и подъёмами, или заданным порядком следования элементов [6-7], частным случаем которых являются альтернативные перестановки [8-9].

Целью работы является решение задач перечисления и генерации обобщенного случая перестановок с заданными спусками и подъёмами: перестановок, где порядок следования элементов определен частично.

1. Перестановки с частично определенным порядком следования элементов

Пусть задано множество порождающих элементов $A=\{a_1,a_2,...,a_n\}$, $a_1<a_2<...<a_n$, $a_i\in R^1$, $i\in J_n$, $J_n=\{1,2,...,n\}$. Множество всевозможных перестановок из элементов $a_1,a_2,...,a_n$ обозначим P_n. Для любой перестановки $\pi=(\pi_1,\pi_2,...,\pi_n)\in P_n$ справедливы соотношения:

$$\pi_1\,\rho_1\,\pi_2\,\rho_2\,...\rho_{n-1}\,\pi_n\,,\ \rho_i\in\{<,>\}\,,\ i\in J_{n-1}\,. \tag{1}$$

Иначе говоря,

$$\pi_i<\pi_{i+1} \tag{2}$$

или

$$\pi_i>\pi_{i+1}\,. \tag{3}$$

Здесь $i\in J_{n-1}$.

Последовательность $\rho(\pi)=(\rho_1,\rho_2,...,\rho_{n-1})$ назовем порядком следования элементов перестановки $\pi\in P_n$.

Лемма 1. Во множестве P_n всевозможных перестановок из n элементов существует 2^{n-1} различных порядков следования элементов $\rho(\pi)$.

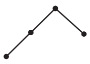
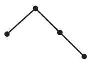
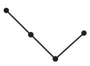
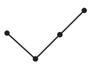
Доказательство леммы следует из существования взаимно однозначного соответствия между множеством всех последовательностей $\rho_1,\rho_2,...,\rho_{n-1}$ и множеством всех двоичных последовательностей из $n-1$ элемента.

Порядок следования элементов $\rho(\pi)$ перестановки $\pi\in P_n$ можно представить с помощью диаграммы, состоящей из $n-1$ сегмента. Выполнению неравенства (2) соответствует участок возрастания, а выполнению неравенства (3) – участок убывания.

Пример 1. Множество перестановок P_4 порождено элементами $\{1,2,3,4\}$. Согласно лемме 1, существует всего 8 различных порядков следования элементов перестановок множества P_4. Диаграммы, представляющие порядки следования, и соответствующие им перестановки приведены в таблице 1.

Таблица 1

Диаграммы и перестановки множества P_4

Диаграмма	Перестановки	Диаграмма	Перестановки
	(1,2,3,4)		(4,3,2,1)
	(1,2,4,3) (1,3,4,2) (2,3,4,1)		(3,4,2,1) (2,4,3,1) (1,4,3,2)
	(3,2,1,4) (4,3,1,2) (4,2,1,3)		(4,1,2,3) (2,1,3,4) (3,1,2,4)
	(1,3,2,4) (1,4,2,3) (2,4,1,3) (2,3,1,4) (3,4,1,2)		(4,2,3,1) (3,2,4,1) (3,1,4,2) (4,1,3,2) (2,1,4,3)

Произвольную подпоследовательность π^i, то есть последовательность $r(\pi)=(\rho_{i_1},\rho_{i_2},\ldots,\rho_{i_k})$, $i_1<i_2<\ldots<i_k$, $k\in J_{n-1}$, назовем частично определенным порядком следования элементов (либо частично определенным порядком следования) перестановки $\pi\in P_n$. «Полный» порядок следования π^i определяет отношение элементов π_i и π_{i+1} для всех позиций $i\in J_{n-1}$, частично заданный порядок $r(\pi)$ — лишь для некоторых $i\in\{i_1,i_2,\ldots,i_k\}\subseteq J_{n-1}$.

Зафиксируем выбранный частично заданный порядок следования элементов $r(\pi)$ перестановки $\pi\in P_n$. Ему можно поставить в соответствие часть множества спуска перестановки. Множество $D(\pi)$ спуска перестановки π определяется как [7]:

$$D(\pi)=\{i\,|\,\pi_i>\pi_{i+1}\},\; i\in J_{n-1}.$$

Определим множество $\bar{D}(\pi)=J_{n-1}\setminus D(\pi)$ как:

$$\bar{D}(\pi)=\{i\,|\,\pi_i<\pi_{i+1}\},\; i\in J_{n-1}.$$

Нас интересует часть $D(\pi)$, касающаяся позиций $i\in\{i_1,i_2,\ldots,i_k\}$. Обозначим её через $D'(\pi)$:

$$D'(\pi)=\{i\,|\,\pi_i>\pi_{i+1}\}\subseteq D(\pi),\; i\in\{i_1,i_2,\ldots,i_k\}.$$

Тогда множество $\bar{D}'(\pi)=\{i_1,i_2,\ldots,i_k\}\setminus D'(\pi)$ можно определить в виде

$$\bar{D}'(\pi)=\{i\,|\,\pi_i<\pi_{i+1}\}\subseteq\bar{D}(\pi),\; i\in\{i_1,i_2,\ldots,i_k\}.$$

Множества $D'(\pi)$ и $\bar{D}'(\pi)$ определяют частично заданный порядок следования $r(\pi)$ элементов перестановки $\pi\in P_n$. Частично заданный порядок следования $r(\pi)$ и множества $D'(\pi)$ и $\bar{D}'(\pi)$ однозначно определяют друг друга. Например, если $r(\pi)=(\rho_2,\rho_4,\rho_5)=(>,<,>)$, то $D'(\pi)=\{2,5\}$ и $\bar{D}'(\pi)=\{4\}$. Это же утверждение справедливо для «полного» порядка следования $\rho(\pi)$ как частного случая $r(\pi)$. Например, в последней строке таблицы 1 множество спуска перестановок, соответствующее диаграмме слева, содержит один элемент: $D(\pi)=\{2\}$, а $\bar{D}(\pi)=\{1,3\}$. Для перестановок, соответствующих диаграмме справа, $D(\pi)=\{1,3\}$, $\bar{D}(\pi)=\{2\}$.

Перестановка удовлетворяет частично заданному порядку $r(\pi)$ тогда, когда для всех $i\in\{i_1,i_2,\ldots,i_k\}$ отношение π_i и π_{i+1} соответствует заданному в $r(\pi)$. Отношение π_i и π_{i+1} для $i\notin\{i_1,i_2,\ldots,i_k\}$ при этом может быть любым:

$$\rho_i\in\{<,>\},\; i\notin\{i_1,i_2,\ldots,i_k\}.$$

Лемма 2. Чтобы перестановка $\pi\in P_n$ удовлетворяла частично заданному порядку следования $r(\pi)$, заданному для k позиций $i\in\{i_1,i_2,\ldots,i_k\}$, необходимо и достаточно, чтобы она удовлетворяла одному из 2^{n-k-1} «полных» порядков следования $\rho(\pi)$, совпадающих с $r(\pi)$ для $i\in\{i_1,i_2,\ldots,i_k\}$.

Доказательство. Так как отношение элементов π_i и π_{i+1} при $i\notin\{i_1,i_2,\ldots,i_k\}$ может быть любым, то в перестановках, удовлетворяющих частично заданному порядку $r(\pi)$, на позициях $i\notin\{i_1,i_2,\ldots,i_k\}$ может иметь место как неравенство $\pi_i<\pi_{i+1}$, так и $\pi_i>\pi_{i+1}$. Это означает, что если частично заданный порядок $r(\pi)$ задан для k позиций из $n-1$ возможных, то для остальных $n-1-k$ позиций $i\notin\{i_1,i_2,\ldots,i_k\}$ отношение соседних элементов может быть любым из 2 возможных ($\rho_i\in\{<,>\}$). Следовательно, для каждой из остальных $n-1-k$ позиций $i\notin\{i_1,i_2,\ldots,i_k\}$ можно зафиксировать любое отношение $\rho_i\in\{<,>\}$, получив тем самым 2^{n-k-1} «полных» порядков следования $\rho(\pi)$. В перестановках, удовлетворяющих каждому такому порядку следования $\rho(\pi)$, отношение π_i и π_{i+1} при $i\in\{i_1,i_2,\ldots,i_k\}$ будет соответствовать порядку, заданному в $r(\pi)$.

Если в частично заданном порядке следования $r(\pi)$ заданы все позиции $i\in\{1,2,\ldots,n-1\}$, перестановки с частично определенным порядком следования превращаются в перестановки с полностью заданным порядком следования $\rho(\pi)$. Частными случаями таких перестановок являются унимодальные перестановки [10] при $\rho(\pi)=(>,>,\ldots,>,<,\ldots,<)$ либо $\rho(\pi)=(<,<,\ldots,<,>,\ldots,>)$ (т.е. когда сначала идут только спуски, а потом — только подъёмы и наоборот), а также альтернативные перестановки [9] при $\rho(\pi)=(>,<,>,<,\ldots)$ либо $\rho(\pi)=(<,>,<,>,\ldots)$ (т.е. когда спуски и подъёмы чередуются).

2. Перечисление перестановок с частично определенным порядком следования элементов

Рассмотрим задачу перечисления перестановок $\pi\in P_n$ с частично определенным порядком следования элементов $r(\pi)$.

Из леммы 2 следует, что количество перестановок, удовлетворяющих частично заданному порядку $r(\pi)$, равно сумме количества перестановок, удовлетворяющих каждому из 2^{n-k-1} «полных» порядков следования $\rho^j(\pi)$, $j=1,2,\ldots,2^{n-k-1}$.

Тогда необходимо для частично заданного порядка следования $r(\pi)$ найти все возможные соответствующие ему «полные» порядки следования $\rho^1(\pi),\rho^2(\pi),...,\rho^{2^{n-k-1}}(\pi)$, затем определить количество перестановок, соответствующих каждому из них, и сложить.

Формулы, полученные в [7], позволяют с помощью принципа включений-исключений определить количество перестановок, удовлетворяющих заданному множеству спуска, т.е. перестановок с заданным «полным» порядком следования.

Рассмотрим задачу перечисления перестановок $\pi \in P_n$ с порядком следования элементов $\rho(\pi)$. Это условие эквивалентно тому, что множеством спуска каждой из перечисляемых перестановок является множество $S \subseteq J_{n-1}$. Другими словами, для перечисления перестановок $\pi \in P_n$ с порядком следования элементов $\rho(\pi)$ необходимо подсчитать число $N(\rho(\pi))$ всех перестановок $\pi \in P_n$, таких, что $D(\pi) = S$.

Следуя [7], для $S = \{s_1, s_2, ..., s_k\} \subseteq J_{n-1}$ обозначим через $\beta_n(S)$ количество перестановок $\pi \in P_n$, для которых S является множеством спуска.

В [7] получена следующая формула для вычисления $\beta_n(S)$:

$$\beta_n(S) = \sum_{1 \le i_1 < i_2 < ... < i_j \le k} (-1)^{k-j} \begin{pmatrix} n \\ s_{i_1}, s_{i_2} - s_{i_1}, ..., n - s_{i_j} \end{pmatrix}, \quad (4)$$

где

$$\begin{pmatrix} n \\ n_1, n_2, ..., n_k \end{pmatrix} = \frac{n!}{n_1! \cdot n_2! \cdot ... \cdot n_k!}.$$

Приведенные рассуждения делают справедливой следующую лемму:

Лемма 3. Пусть $\rho(\pi^0)$ – порядок следования элементов перестановки $\pi^0 \in P_n$, причем $D(\pi^0) = S = \{s_1, s_2, ..., s_k\} \subseteq J_{n-1}$. Количество $N(\rho(\pi^0))$ всех перестановок $\pi \in P_n$, имеющих порядок следования элементов $\rho(\pi) = \rho(\pi^0)$, равно $\beta_n(S)$, т.е. $N(\rho(\pi^0)) = \beta_n(S)$, и определяется по формуле (4).

Доказательство. Определим бинарное отношение R во множестве P_n следующим образом. Два элемента $\pi^1, \pi^2 \in P_n$ находятся в отношении R, если они имеют один и тот же порядок следования элементов: $\rho(\pi^1) = \rho(\pi^2)$. Для отношения R нетрудно проверить справедливость свойств рефлексивности, симметричности и транзитивности, что позволяет сделать вывод о том, что R является отношением эквивалентности во множестве P_n.

Тогда все элементы $\pi \in P_n$, имеющие один и тот же порядок следования элементов, образуют класс эквивалентности $[\pi]$ по отношению R. Множество всех классов эквивалентности составляет фактор-множество $\Pi = P_n / R$ множества P_n по отношению R.

Пусть $\Lambda : P_n \to \Pi$ – фактор-отображение на множестве P_n. Рассмотрим задачу реализации фактор-отображения Λ в явном виде. Решение этой задачи сводится к генерации всех перестановок $\pi \in P_n$ с порядком следования элементов $\rho(\pi) = \rho(\pi^0)$ для любой перестановки $\pi^0 \in P_n$.

Сложив количество перестановок, имеющих порядок следования $\rho^1(\pi),\rho^2(\pi),...,\rho^{2^{n-k-1}}(\pi)$, получим количество перестановок, удовлетворяющих заданному частично заданному порядку $r(\pi)$:

$$N(r(\pi)) = \sum_{j=1}^{2^{n-k-1}} N(\rho^j(\pi)) \quad (5)$$

Построим алгоритм генерации перестановок $\pi \in P_n$ с частично определенным порядком следования элементов $r(\pi)$.

3. Генерация перестановок с частично определенным порядком следования элементов

Пусть заданы множество порождающих элементов $A = \{a_1, a_2, ..., a_n\}$, $a_1 < a_2 < ... < a_n$ и частично заданный порядок следования элементов $r(\pi)$, определяющий множества $D'(\pi)$ и $\overline{D}'(\pi)$. Необходимо сгенерировать все перестановки $\pi \in P_n$, соответствующие $D'(\pi)$ и $\overline{D}'(\pi)$, в количестве $N(r(\pi))$, определяемом по формуле (5).

Обозначим через π^i частичную перестановку, состоящую из i элементов, $\pi^i = (\pi_1, \pi_2, ..., \pi_i)$, $\pi_i \in A$, $i \in J_n^0$, $J_n^0 = \{0,1,2,...,n\}$. При этом π^0 – пустая частичная перестановка, $\pi^n = \pi \in P_n$ – результат генерации. Опишем работу алгоритма генерации перестановок с заданным порядком следования элементов $r(\pi)$. Назовём его *PartOrderedPerm.*

В начале работы, на нулевом уровне, входом алгоритма *PartOrderedPerm* является пустая перестановка π^0.

Алгоритм *PartOrderedPerm* работает рекурсивно, на каждом i-ом уровне рекурсии $i \in J_{n-1}^0$ добавляя в текущую частичную перестановку $\pi^i = (\pi_1, \pi_2, ..., \pi_i)$ длины i следующий элемент с номером $i+1$, тем самым получая на следующем уровне частичную перестановку $\pi^{i+1} = (\pi_1, \pi_2, ..., \pi_{i+1})$ длины $i+1$. На уровне n алгоритм выводит .

Элемент $\pi_{i+1} \in A$ должен удовлетворять определенным условиям. Обозначим через F^i множество всех порождающих элементов, удовлетворяющих этим условиям, на каждом уровне $i \in J_{n-1}^0$.

Также обозначим через m^i *количество следующих подряд подъёмов либо спусков в частично заданном порядке следования, начиная со следующей позиции* $i+1$. Посчитаем значения m^i на каждом уровне $i \in J_{n-2}^0$ с помощью алгоритма *Get_M*, приведенного на рис.1.

Полученное множество значений $\{m_0, m_1, ..., m_{n-2}\}$ является входным параметром алгоритма *Get_F* для получения F^i, приведенного на рис.2.

Опишем теперь основной алгоритм *PartOrderedPerm*, генерирующий все перестановки с заданным порядком следования элементов. Этот алгоритм приведен на рисунке 3. Для генерации всех необходимых перестановок *PartOrderedPerm* вызывается с параметром $\pi^0 = ()$.

```
def Get_M(i, n, D'(π), D̄'(π))
for i in {0, 1, ..., n - 2}
    if {i+1} ⊆ D'(π)
        m^i := max(t : i + t ∈ D'(π))
    else
        m^i := max(t : i + t ∈ D̄'(π))
```

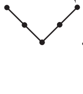
Рис. 1. Алгоритм *Get_M*

```
def Get_F(i, n, D'(π), D̄'(π), {a_1, a_2, ..., a_n}, {m_0, m_1, ..., m_{n-2}}, π^i);
F^i := ∅
if i = 0
    if 1 ∈ D'(π)
        return F^i := {a_{m^0+1}, a_{m^0+2}, ..., a_n}
    if 1 ∈ D̄'(π)
        return F^i := {a_1, a_2, ..., a_{n-m^0}}
    if 1 ∉ D'(π), 1 ∉ D̄'(π)
        return F^i := {a_1, a_2, ..., a_n}
for (a ∈ A, a ≠ π^i_j, j = 1, 2, ..., i)
    if (i ∈ D(π) and a < π^i_i) or (i ∈ D̄'(π) and a > π^i_i) or i ∉ D'(π), i ∉ D̄'(π)
        condition2:=true else condition2:=false
    if i+1 ∈ D'(π)
        condition3:= ∃L = {a_{l_1}, a_{l_2}, ..., a_{l_{m^i}}} ⊆ A : ∀a_{l_j} < π_{i+1}, a_{l_j} ≠ π_k,
                    ∀j ∈ J_{m^i}, ∀k ∈ J_i
    if i+1 ∈ D̄'(π)
        condition3:= ∃L = {a_{l_1}, a_{l_2}, ..., a_{l_{m^i}}} ⊆ A : ∀a_{l_j} > π_{i+1}, a_{l_j} ≠ π_k,
                    ∀j ∈ J_{m^i}, ∀k ∈ J_i
    if i+1 ∉ D'(π), i+1 ∉ D̄'(π)
        condition3:=true
    if (i = n−1) or (condition2 and condition3)
        F^i := F^i ∪ a
return F^i
```

Рис. 2. Алгоритм формирования множества

```
def PartOrderedPerm (i, n, D'(π), D̄'(π), {a_1, a_2, ..., a_n}, {m_0, m_1, ..., m_{n-2}}, π^i)
if i = n
    # we have got a full permutation, output it
    output π^i
    exit
F^i := Get_F(i, n, D'(π), D̄'(π), {a_1, a_2, ..., a_n}, {m_0, m_1, ..., m_{n-2}}, π^i);
for f^i_j in F^i
    PartOrderedPerm(i, n, D'(π), D̄'(π), {a_1, a_2, ..., a_n}, {m_0, m_1, ..., m_{n-2}},
                    π^{i+1} = (π_1, π_2, ..., π_i, f^i_j))
```

Рис. 3. Основной алгоритм *PartOrderedPerm*

Пример 2. На рис. 4 представлен пример генерации перестановок из элементов $A = \{1,2,3,4,5\}$ при заданных множествах $D'(\pi) = \{1\}, \overline{D}'(\pi) = \{3,4\}$ (позиция $i = 2$ не задана). Лемма 2 в данном случае звучит так: чтобы перестановка $\pi \in P_n$ удовлетворяла частично заданному порядку следования $r(\pi)$, заданному для $k = 3$ позиций $i \in \{1,3,4\}$, необходимо и достаточно, чтобы она удовлетворяла одному из 2 «полных» порядков следования $\rho(\pi)$, совпадающих с $r(\pi)$ для $i \in \{1,3,4\}$. Такие «полные» порядки следования должны иметь спуск на позиции 1 и подъём на позициях 3,4 и соответствуют диаграммам и .

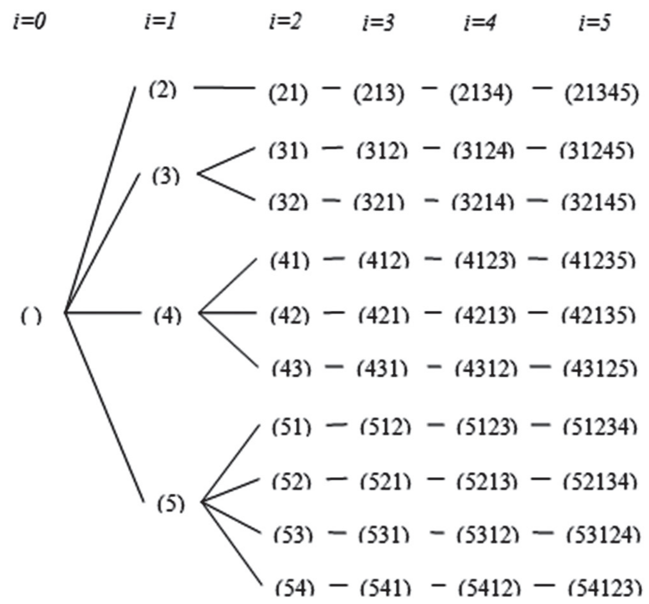


Рис. 4. Рекурсивное дерево работы алгоритма при генерации перестановок из элементов $\{1,2,3,4,5\}$ и множествах $D'(\pi) = \{1\}, \overline{D}'(\pi) = \{3,4\}$

Применим алгоритм *Get_M* для подсчёта значений m^i. В результате получим $m^0 = 1$, m^1 не считается (позиция 2 не задана), $m^2 = 2$, $m^3 = 1$.

Приведём множества F^i на уровнях $i = 0,1$. На уровне $i = 0$ имеем $\pi^0 = (\ )$, $1 \in D'(\pi)$ и $F^0 = \{a_{1+1}, \ldots, a_5\} = \{2,3,4,5\}$.

Так как множество F^0 содержит 4 элемента, алгоритм «разветвляется», имея на уровне $i = 1$ частичные перестановки $\pi^1 = (2), \pi^1 = (3), \pi^1 = (4), \pi^1 = (5)$.

Рассмотрим пример формирования множества F^1 для частичной перестановки $\pi^1 = (3)$. Порождающими элементами, которые удовлетворяют вышеописанным условиям и могут быть на месте π_2, являются элементы $\{1,2\}$.На этом же уровне для перестановки $\pi^1 = (4)$ множество $F^1 = \{1,2,3\}$.

Так как для $\pi^1 = (1)$ множество F^1 содержит 2 элемента, то снова происходит разветвление рекурсивного дерева. На уровнях $i = 2,3,4$ множества F^2, F^3 и F^4 содержат по одному элементу, поэтому разветвлений нет. На уровне $i = 5$ длина перестановки равна 5, алгоритм выводит $\pi = \pi^5$ на экран.

Посчитав количество необходимых перестановок по формуле (5), получим в результате 10, что совпадает с количеством перестановок, полученных в результате работы алгоритма.

Выводы

В работе описан специальный вид перестановок — перестановки с частично определенным порядком следования. Оценено количество таких перестановок, предложен алгоритм для их генерации. Отметим, что описанный алгоритм *PartOrderedPerm* является универсальным и может быть использован для генерации других комбинаторных множеств. Для этого достаточно задать правила формирования множества для конкретного комбинаторного множества. Частными случаями описанного класса перестановок являются унимодальные и альтернативные перестановки.

Список литературы: 1. *Ruskey F.* Combinatorial generation [Text] / F. Ruskey. – Dept. of Computer Science Univ. of Victoria, Canada, 2003.– 289 p. **2.** *Kreher D.* Combinatorial Algorithms: Generation Enumeration and Search [Text] / D. Kreher, D. Stinson. – CRC Press, 1999. – 329 p. **3.** *Knuth D.* The Art of Computer Programming [Text] / D.Knuth. – Volume 4, Fascicle 2: Generating All Tuples and Permutations. – Addison-Wesley, 2005. **4.** *Гребенник, И.В.* Описание и генерация перестановок, содержащих циклы [Текст] / И.В. Гребенник // Кибернетика и системный анализ. – 2010. – № 6. – С. 97-105. **5.** *Poneti M.* Generating restricted classes of involutions, Bell and Stirling permutations [Text] / M. Poneti, V. Vajnovszki. – European Journal of Combinatorics.–2010. – №31. – pp. 553–564. **6.** *van Baronaigien R.* Generating permutations with given ups and downs [Text] / R. van Baronaigien, F. Ruskey // Discrete Applied Mathematics. – № 36(1), 1992. – pp. 57–65. **7.** *Stanley R.* Enumerative combinatorics [Text] / R.P. Stanley. –Wadsworth, Inc. California. – vol. 1 – 1986. **8.** *Stanley R. P.* A survey of alternating permutations // Contemp. Math. – 2010. – Т. 531. – С. 165-196. **9.** *Bauslaugh B.,* Ruskey F. Generating alternating permutations lexicographically // BIT Numerical Mathematics. – 1990. – Т. 30. – №. 1. – С. 17-26. **10.** *Bergeron F.,* Labelle G., Leroux P. Combinatorial species and tree-like structures. – Cambridge University Press, 1998. – Т. 67. **11.** *Grebennik I. V.,* Lytvynenko O. S. Generating combinatorial sets with given properties // Cybernetics and Systems Analysis. – 2012. – С. 1-9.

Resume

I.V. Grebennik, O.S. Lytvynenko

Generating permutations with partially defined order of elements

Special class of permutations is considered– permutations with partially defined order of elements. Such class of permutations is a generalization of per mutations with given ups and downs. A formula for enumeration of such permutations as well and generation algorithm is given. Described algorithm can be used for generation of other combinatorial sets.

Background. It is sometimes insufficient to use classical combinatorial sets (arrangements, permutations, combinations) for constructing mathematical models of some combinatorial optimization problems. In such cases, non-classical combinatorial sets are used, and one of such sets are permutations with given ups and downs or permutations with defined order of elements. However, a problem of enumerating and generating permutations, where order or elements is defined partially, has not solved yet.

Materials and methods. The inclusion-exclusion principle was applied for enumerating permutations with partially defined order of elements. For generating all permutations with partially defined order of elements, recursive backtracking algorithm was used.

Results. Mathematical description for permutations with partially defined order of elements was given. A formula for enumerating all such permutations was constructed. Also article describes recursive backtracking algorithm for generating all permutations with partially defined order of elements.

Conclusion. Described algorithm for generating permutations with partially defined order of elements is quite universal and can be used for generating other classes of combinatorial sets.

Поступила в редколлегию 4.09.2017.