КОМПЬЮТЕРНАЯ МАТЕМАТИКА

УДК 519.115

С.В. ЧУПОВ

ЛЕКСИКОГРАФІЧНО ВПОРЯДКОВАНІ ПЕРЕСТАНОВКИ

Представлено альтернативний спосіб запису перестановок, який названо позиційним представленням перестановки. На множині лексикографічно впорядкованих позиційних представлень перестановок, формулюються різноманітні алгебраїчні операції над перестановками у їх позиційному представленні. Доводиться, що між операціями додавання та множення на множині всіх позиційних представлень перестановок n-го порядку, операціями суми та добутку по модулю n! існує ізоморфне відображення.

Вступ. Перестановки відіграють важливу роль у математиці та інформатиці. Зокрема в задачах комбінаторики, дискретної математики, багатьох оптимізаційних задачах [1 – 3]. Так в класичній задачі про комівояжера та квадратичній задачі про призначення перестановки є прямим об'єктом дослідження. Як вказувалося в [4], від вдалого вибору лексикографічного впорядкування змінних задачі залежить ефективність роботи точних та наближених методів лексикографічного напрямленого перебору. Чим ближче в лексикографічному розумінні буде розташований оптимальний розв'язок задачі в даному лексикографічному порядку до лексикографічного екстремуму множини допустимих значень тим швидше він буде отриманий. Але кожному лексикографічному порядку відповідає певна перестановка індексів змінних. Тому проблема перегляду, аналізу та перетворення перестановок у процесі розв'язання багатьох задач, зокрема задач дискретної оптимізації є дуже важлива та актуальна. Але не слід забувати, що навіть при відносно невеликій розмірності перестановки їх загальна кількість є великою. Так, наприклад, кількість перестановок розмірності 50 представляє собою 65-и розрядне ціле десяткове число:

$$50! = 30414093201713378043612608166064768844377641568960512000000000000.$$

В роботі розглядається альтернативний спосіб представлення перестановок, який дозволяє над впорядкованими перестановками визначати різноманітні алгебраїчні операції.

Позиційне представлення перестановок. Якщо всі перестановки однієї розмірності впорядкувати за лексикографічним зростанням, тоді можна помітити певну закономірність у їх розташуванні, яка дозволяє використовувати замість звичайного представлення перестановок інший вигляд. На рис. 1 показано всі перестановки розмірності 4, які впорядковано в порядку лексикографічного зростання. Останній рядок на цьому рисунку містить порядкові номери або індекси перестановок у даному наборі.

π	0	0	0	0	0	0	1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3
	1	1	2	2	3	3	0	0	2	2	3	3	0	0	1	1	3	3	0	0	1	1	2	2
	2	3	1	3	1	2	2	3	0	3	0	2	1	3	0	3	0	1	1	2	0	2	0	1
	3	2	3	1	2	1	3	2	3	0	2	0	3	1	3	0	1	0	2	1	2	0	1	0
№	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

РИС. 1. Лексикографічно впорядковані перестановки розмірності 4

Якщо скористатися цією закономірністю та узагальнити її для перестановок довільної розмірності *n*, тоді перестановку π можна записати так, що в результаті отримаємо цілочисловий вектор *p*, але на відміну від звичайної перестановки цей вектор буде складатися з позицій або індексів, що визначаються певним чином, яким відповідають елементи перестановки π. На рис. 2 показано структурну схему такого перетворення. Рис. 3 містить структурну схему алгоритму для оберненого перетворення до перестановки звичайного виду. Зауважимо, що нумерація елементів у перестановках π, Ord та p починається від 0.

$$p = PermutationToPosition(\pi, Ord)$$

Input: π – перестановка,
Ord – базова перестановка
Output: *p* – позиційне представлення

for(int i = 0; i < n - 1; i++){
$p_i = Index\ of\ \pi_i\ in\ Ord;$
$Ord = Ord \setminus \{\pi_i\};$
}
$p_{n-1} = 0;$

РИС. 2. Алгоритм перетворення перестановки до позиційного виду

$$\pi = PositionToPermutation(p, Ord)$$

Input: *p* – позиційне представлення,
Ord – базова перестановка
Output: π – перестановка

for(int i = 0; i < n; i++){
$\pi_i = Ord_{p_i};$
$Ord = Ord \setminus \{\pi_i\};$
}

РИС. 3. Алгоритм перетворення позиційного представлення перестановки до стандартного виду

Означення 1. Вектор цілих чисел $p = (p_0, p_1, ..., p_{n-1})$, отриманий в результаті роботи алгоритму $p = PermutationToPosition(\pi, Ord)$ назвемо позиційним представленням перестановки $\pi = (\pi_0, \pi_1, ..., \pi_{n-1})$.

Легко бачити, що довільний цілочисловий вектор $p = (p_0, p_1, ..., p_{n-1})$ у якого кожна координата $p_i \in \{0,1,...,n-i-1\}$, $i = 0,1,...,n-1$ є позиційним представленням деякої перестановки π. Надалі мінімальне ціле значення, яке більше за p_i позначатимемо $power_i$, тобто $power_i = n - i$, $i = 0,1,...,n-1$.

Базова перестановка *Ord* в алгоритмах *PermutationToPosition* та *PositionToPermutation* визначає яким чином будуть впорядковані перестановки. При цьому впорядкований набір перестановок завжди буде починатися з базової перестановки *Ord*. Якщо $Ord = (0,1,...,n-1)$ – тривіальна перестановка, тоді за вищерозглянутими алгоритмами отримаємо стандартне лексикографічне впорядкування перестановок за зростанням, як це показано на рис. 1. Якщо, наприклад, $Ord = (1,3,0,2)$, тоді за алгоритми *PermutationToPosition* та *PositionToPermutation* отримаємо впорядкування перестановок, яке показано на рис. 4.

	1	1	1	1	1	1	3	3	3	3	3	3	0	0	0	0	0	0	2	2	2	2	2	2
	3	3	0	0	2	2	1	1	0	0	2	2	1	1	3	3	2	2	1	1	3	3	0	0
π	0	2	3	2	3	0	0	2	1	2	1	0	3	2	1	2	1	3	3	0	1	0	1	3
	2	0	2	3	0	3	2	0	2	1	0	1	2	3	2	1	3	1	0	3	0	1	3	1
№	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

РИС. 4. Впорядкування перестановок для $Ord = (1,3,0,2)$

На рис. 5 показано всі позиційні представлення перестановок отримані за алгоритмом *PermutationToPosition*, що відповідають перестановкам, які показані на рис. 1, якщо базова перестановка *Ord* є тривіальною. Той самий впорядкований набір позиційних представлень отримаємо, якщо використовувати впорядкований набір перестановок з рис. 4. Взагалі, варто зазначити, що для довільної базової перестановки *Ord* та відповідного їй впорядкованого набору перестановок завжди має місце стандартне лексикографічне впорядкування позиційних представлень перестановок, які отримуються в результаті застосування алгоритму *PermutationToPosition*. Крім того, не важко помітити, що значення останнього елемента в довільному позиційному представленні завжди дорівнює 0.

	0	0	0	0	0	0	1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3
p	0	0	1	1	2	2	0	0	1	1	2	2	0	0	1	1	2	2	0	0	1	1	2	2
	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
№	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

РИС. 5. Позиційне представлення перестановок розмірності 4

Між множиною усіх перестановок розмірності *n* та множиною позиційних представлень розмірності *n* існує взаємно однозначна відповідність.

Теорема 1. Для довільної перестановки $\pi = (\pi_0, \pi_1, ..., \pi_{n-1})$ та базової перестановки *Ord* існує одне і тільки одне позиційне представлення $p = (p_0, p_1, ..., p_{n-1})$, що визначається алгоритмами *PermutationToPosition* та *PositionToPermutation*.

Доведення. В індексованому наборі елементів будь-який елемент однозначно визначається своїм індексом. Крім того в довільній перестановці π усі елементи різні. Тому на кожному *i*-му кроці алгоритму *PermutationToPosition* елемента перестановки π_i відповідає одна і тільки одна позиція p_i у базовій перестановці *Ord*.

Нехай маємо вектор $p = (p_0, p_1, ..., p_{n-1})$, де $p_i \in \{x \in Z \mid 0 \le x < n - i\}$, $i = 0,1,...,n-1$. Покажемо, що *p* однозначно визначає певну перестановку $\pi = (\pi_0, \pi_1, ..., \pi_{n-1})$. Зауважимо, що на початку роботи алгоритму *PositionToPermutation* впорядкований набір *Ord* представляє собою довільну перестановку розмірності *n*. На кожному *i*-му кроці алгоритму $0 \le p_i < |Ord|$ та *Ord* є перестановкою розмірності $n - i$, тому для кожного Ord_{p_i} завжди існує лише одне значення p_i, крім того $\{\pi_0, \pi_1, ..., \pi_{i-1}\} \cap Ord = \varnothing$, де елементи множини $\{\pi_0, \pi_1, ..., \pi_{i-1}\}$ складають перестановку розмірності *i*. Здійснивши усі *n* кроків за алгоритмом *PositionToPermutation* отримаємо перестановку π розмірності *n*.

Алгебраїчні операції над перестановками в позиційному представленні. Виявляється, що якщо перестановки розглядати в позиційному виді, тоді з'являється можливість проводити з ними різноманітні алгебраїчні дії так, що результуюча перестановка завжди буде займати правильне місце серед усіх перестановок, що впорядковано за базовою перестановкою *Ord*. В наступних структурних схемах використовуються такі позначення: $a \bmod b$ – остача від ділення цілого *a* на ціле *b*; $a \operatorname{div} b$ – цілочислове ділення цілого *a* на ціле *b*.

На рис. 6 та 7 показано структурні схеми алгоритмів для відшукання суми та різниці двох позиційних представлень перестановок однієї розмірності відповідно. На кожному кроці *k* цілочислове значення *power* представляє собою верхню межу значень, яка обмежує значення p_k у даній позиції ($p_k < power$). Цілочислове значення *carry* – величина переносу, яка враховується на наступному кроці.

Варто зазначити, якщо після завершення роботи алгоритму додавання значення переносу буде відмінним від нуля, то це буде означати, що результат матиме більшу розмірність ніж його операнди. При нехтуванні додатного переносу отримаємо циклічний перехід на відповідне місце з початку в множині лексикографічно впорядкованих позиційних представлень. Так, якщо знаходити (3, 2, 0, 0) + (0, 1, 1, 0), значення переносу в кінці буде рівним 1 і при його нехтуванні в результаті отримаємо позиційне представлення (0, 0, 1, 0).

Нехай за алгоритмом *AddPositions* виконано операцію додавання $p = p^1 + p^2$. На кожному кроці k алгоритму додавання значення результуючого розряду p_k та розрядів операндів p_k^1 і p_k^2 зв'язані співвідношенням: $p_k^1 + p_k^2 + carry_{k+1} = carry_k * power_k + p_k$, де $carry_{k+1}$ – значення переносу з попереднього розряду, $carry_k$ – перенос з поточного розряду. Припустимо, що відомий результат додавання p та один з його доданків p^2. Виразимо p_k^1 та отримаємо: $p_k^1 = p_k - p_k^2 + carry_k * power_k - carry_{k+1}$. Легко показати, що для довільного $k = 0,1,...,n-2$, $carry_k \in \{0,1\}$. Зауважимо, що процес визначення p_k^1, $k = 1,2,...,n-1$ є операцією знаходження різниці. Для визначення p_k^1 невідоме лише значення $carry_k$.

$$p = AddPositions(p^1, p^2)$$

Input: p^1, p^2 – позиційні представлення перестановок

Output: позиційне представлення $p = p^1 + p^2$.

$$\begin{aligned}
& p_{n-1} = 0; \\
& int\ carry = 0; \\
& for\left(int\ k = n-2; k \ge 0; k--\right)\{ \\
& \quad int\ summ = p_k^1 + p_k^2; \\
& \quad int\ power = n - k; \\
& \quad p_k = \left(summ + carry\right) mod\ power; \\
& \quad carry = \left(summ + carry\right) div\ power; \\
& \}
\end{aligned}$$

РИС. 6. Алгоритм додавання двох позиційних представлень

$$p = SubPositions(p^1, p^2)$$

Input: p^1, p^2 – позиційні представлення перестановок

Output: позиційне представлення $p = p^1 - p^2$

$$\begin{aligned}
& p_{n-1} = 0; \\
& int\ carry = 0; \\
& for\left(int\ k = n-2; k \ge 0; k--\right)\{ \\
& \quad int\ power = n - k; \\
& \quad int\ newCarry = 0; \\
& \quad int\ pk = p_k^1 - p_k^2 - carry; \\
& \quad if\ (pk < 0)\ newCarry = 1; \\
& \quad p_k = pk + newCarry * power; \\
& \quad carry = newCarry; \\
& \}
\end{aligned}$$

РИС. 7. Алгоритм відшукання різниці двох позиційних представлень

Але проаналізувавши можливі чотири випадки:

1) $carry_k = 0, carry_{k+1} = 0$, $p_k^1 = p_k - p_k^2$;
2) $carry_k = 0, carry_{k+1} = 1$, $p_k^1 = p_k - p_k^2 - 1$;
3) $carry_k = 1, carry_{k+1} = 0$, $p_k^1 = p_k - p_k^2 + power_k$;
4) $carry_k = 1, carry_{k+1} = 1$, $p_k^1 = p_k - p_k^2 - 1 + power_k$

можна зробити висновок, що запозичення робимо лише тоді ($carry_k = 1$), коли $p_k - p_k^2 - carry_{k+1} < 0$. Таким чином на кожному кроці k значення p_k^1 обчислюється однозначно. Тому для відшукання значення різниці поточного розряду можна використовувати формулу $p_k^1 = p_k - p_k^2 + carry_k * power_k$. Отже якщо $p = p^1 + p^2$, тоді $p^1 = p - p^2$ або $p^2 = p - p^1$ та навпаки. Має місце теорема:

Теорема 2. Для довільних двох позиційних представлень p^1 та p^2 сума $p = p^1 + p^2$ знайдена за алгоритмом *AddPositions* правильна тоді і тільки тоді, коли за алгоритмом SubPositions різниці $p^1 = p - p^2$ або $p^2 = p - p^1$ знайдені правильно.

На рис. 8 та 9 показано структурні схеми порозрядного множення та ділення позиційного представлення перестановки на ціле число відповідно. Зауважимо, якщо після завершення роботи алгоритму множення *carry*>0, тоді результат матиме більшу розмірність, ніж його операнд p^1.

$$p = MulPositions(p^1, multiplier)$$

Input: p^1 – позиційне представлення перестановки;
multiplier – цілий множник

Output: позиційне представлення $p = multiplier * p^1$

$$p_{n-1} = 0;$$
$$int\ carry = 0;$$
$$\begin{bmatrix} for(int\ k = n-2;\ k \ge 0;\ k--)\{ \\ \quad int\ power = n-k; \\ \quad int\ mul = multiplier * p_k^1 + carry; \\ \quad p_k = mul\ mod\ power; \\ \quad carry = mul\ div\ power; \\ \} \end{bmatrix}$$

РИС. 8. Алгоритм множення позиційного представлення на ціле число

$$p = DivPositions(p^1, divider)$$

Input: p^1 – позиційне представлення перестановки;
divider – натуральний дільник

Output: позиційне представлення $p = \frac{p^1}{divider}$

$$p_{n-1} = 0;$$
$$int\ carry = 0;$$
$$\begin{bmatrix} for(int\ k = 0;\ k < n-1;\ k++)\{ \\ \quad int\ power = n-k; \\ \quad p_k = (p_k^1 + power * carry)\ div\ divider; \\ \quad carry = ((p_k^1 + power * carry)\ mod\ divider); \\ \} \end{bmatrix}$$

РИС. 9. Алгоритм ділення позиційного представлення на ціле число

Зазначимо, що якщо після завершення роботи алгоритму ділення значення переносу $carry > 0$, тоді ділення відбулося з остачею. В цьому разі $p^2 \neq s * \left(\frac{p^2}{s} \right)$. Як буде показано в наступному пункті позиційне представлення розмірності n виду: $\overline{1} = \left(\underbrace{0,0,...,0}_{n-2},1,0 \right)$ відіграє роль одиниці. Тоді за наявності додатної остачі rem отримаємо $p^2 = s * \left(\frac{p^2}{s} \right) + rem * \hat{p}$.

Властивості лексикографічно впорядкованих позиційних представлень перестановок. Серед усіх лексикографічно впорядкованих позиційних представлень розмірності n можна виділити два представлення, які мають цілком визначені структуру та зміст.

Означення 2. Позиційне представлення $\left(\underbrace{0,0,...,0}_{n} \right)$ називатимемо позиційним представленням нуля і позначатимемо $\overline{0}$.

Означення 3. Позиційне представлення $\left(\underbrace{0,0,...,0}_{n-2},1,0 \right)$ будемо називати позиційним представленням одиниці і позначатимемо $\overline{1}$.

Не важко помітити, наприклад з рис. 5, що це перші два позиційні представлення у лексикографічно впорядкованому наборі позиційних представлень перестановок.

Легко показати, що для позиційного представлення нуль мають місце наступні властивості: $p + \overline{0} = p$, $p - \overline{0} = p$, $d * \overline{0} = \overline{0}$, $p - p = \overline{0}$, де p – довільне позиційне представлення, d – ціле число.

Нехай маємо два позиційні представлення p^1 та p^2, причому вони розміщені строго лексикографічно один за одним, іншими словами вони є сусідніми та $p^1 >^L p^2$. Тоді $p^1 + \overline{1} = p^2$, $p^2 - p^1 = \overline{1}$.

Оскільки всі позиційні представлення впорядковано лексикографічно, то для довільних двох різних позиційних представлень дуже просто визначається яке з них лексикографічно більше за інше. Але питання на скільки одне позиційне представлення більше або менше за інше лишається відкритим. Для того щоб це визначити розпишемо процес множення позиційного представлення на число за алгоритмом з рис. 8.

$$\begin{cases} d * p_{n-2}^2 = carry_{n-2} * power_{n-2} + p_{n-2}^1 \\ d * p_{n-3}^2 + carry_{n-2} = carry_{n-3} * power_{n-3} + p_{n-3}^1 \\ \dots \\ d * p_k^2 + carry_{k+1} = carry_k * power_k + p_k^1 \\ \dots \\ d * p_0^2 + carry_1 = p_0^1. \end{cases} \quad (1)$$

Система (1) представляє собою порозрядне співвідношення між операндом p^2, результатом p^1 та значеннями переносів у процесі роботи алгоритму множення p^2 на ціле *d*.

В останньому рівнянні системи вважатимемо, що $carry_0 = 0$, тобто останнім переносом при виконанні операції множення знехтуємо. Починаючи з останнього рівняння системи (1) поступово виразимо невідомі $carry_k$ – значення переносів на кожному *k*-му кроці алгоритму множення. В результаті отримаємо:

$$carry_k = \sum_{j=0}^{k-1} p_j^1 \prod_{i=j+1}^{k-1} power_i - d \sum_{j=0}^{k-1} p_j^2 \prod_{i=j+1}^{k-1} power_i \text{ , } k = 1, 2, ..., n-1 \text{ .}$$

Оскільки на початку операції множення переносу не має, $carry_{n-1} = 0$, то отримаємо: $d\sum_{j=0}^{n-2} p_j^2 \prod_{i=j+1}^{n-2} power_i = \sum_{j=0}^{n-2} p_j^1 \prod_{i=j+1}^{n-2} power_i$. З того, що $power_i = n - i$, легко бачити: $\prod_{i=j+1}^{n-2} power_i = (n-i-1)!$ та $d\sum_{j=0}^{n-2} p_j^2 (n-j-1)! = \sum_{j=0}^{n-2} p_j^1 (n-j-1)!$.

Теорема 3. Якщо p^1 та p^2 позиційні представлення, причому $p^1 >^L p^2 >^L \overline{0}$, тоді їх відношення $d = \frac{p^1}{p^2}$ визначається як $d = \frac{\sum_{j=0}^{n-2} p_j^1 (n-j-1)!}{\sum_{j=0}^{n-2} p_j^2 (n-j-1)!}$.

Фактично, знайдене ціле значення відношення двох позиційних представлень визначає кількість лексикографічних інтервалів $\overline{0} \le^L p \le^L p^2$, що містяться у лексикографічному інтервалі $\overline{0} \le^L p \le^L p^1$.

На підставі теореми 3, легко доводяться наступні наслідки.

Наслідок 1. Для довільного позиційного представлення *p* його порядковий номер $IndexOf(p)$ у лексикографічно впорядкованому наборі усіх позиційних представлень визначається як $IndexOf(p) = \sum_{j=0}^{n-2} p_j (n-j-1)!$.

Наслідок 2. Для довільного невід'ємного цілого числа $Index < n!$, існує одне і тільки одне позиційне представлення розмірності n.

Наслідок 3. Для довільного невід'ємного цілого числа $d<n!$, позиційного представлення p має місце: $p * d = p * p^d$, де p^d – позиційне представлення з індексом d у лексикографічно впорядкованій множині позиційних представлень.

Між алгебраїчними операціями над позиційним представленнями перестановки та арифметичними операціями над їх порядковими номерами існує взаємно однозначна відповідність.

Теорема 4. Нехай p^1 та p^2 позиційні представлення, m, d – натуральні числа. $p^{sum} = p^1 + p^2$, $p^{sub} = p^1 - p^2$, $p^{mul} = m * p^1$, $p^{div} = \dfrac{p^1}{d}$ тоді і тільки тоді коли

$$IndexOf\left(p^{sum}\right) = IndexOf\left(p^1\right) + IndexOf\left(p^2\right), IndexOf\left(p^{sub}\right) = IndexOf\left(p^1\right) -$$

$$- IndexOf\left(p^2\right), IndexOf\left(p^{mul}\right) = m * IndexOf\left(p^1\right), \ IndexOf\left(p^{div}\right) = \frac{1}{d} * IndexOf\left(p^1\right).$$

Доведення. Доведення проведемо тільки для операції додавання. Для інших операцій доведення здійснюється аналогічно. Нехай $p^{sum} = p^1 + p^2$. Покажемо, що $IndexOf\left(p^{sum}\right) = IndexOf\left(p^1\right) + IndexOf\left(p^2\right)$. Згідно алгоритму додавання на (див. рис. 6) на k-му кроці $p_k^1 + p_k^2 = carry_{k+1} = p_k^{sum} + carry_k * power_k = p_k^{sum} + carry_k * (n-k)$.

Звідки маємо,

$$IndexOf\left(p^{sum}\right) = \sum_{j=0}^{n-2} p_j^{sum}(n-j-1)! = \sum_{j=0}^{n-2}\left(p_j^1 + p_j^2\right)(n-j-1)! + \sum_{j=0}^{n-2} carry_{j+1} * (n-j-1)! -$$

$- \sum_{j=0}^{n-2} carry_j * (n-j)!$. Оскільки переноси $carry_{n-1}$ та $carry_0$ рівні нулю, отримаємо

$$\sum_{j=0}^{n-3} carry_{j+1} * (n-j-1)! - \sum_{j=1}^{n-2} carry_j * (n-j)! = 0.$$

Отже

$$\sum_{j=0}^{n-2}\left(p_j^1 + p_j^2\right)(n-j-1)! + \sum_{j=0}^{n-2} carry_{j+1} * (n-j-1)! -$$

$$-\sum_{j=0}^{n-2} carry_j * (n-j)! = \sum_{j=0}^{n-2}\left(p_j^1 + p_j^2\right)(n-j-1)! = \sum_{j=0}^{n-2} p^1{}_j (n-j-1)! + \sum_{j=0}^{n-2} p^2{}_j (n-j-1)! =$$

$$= IndexOf\left(p^1\right) + IndexOf\left(p^2\right).$$

Нехай $IndexOf\left(p^{sum}\right)=IndexOf\left(p^{1}\right)+IndexOf\left(p^{2}\right)$. Покажемо, що $p^{sum}=p^{1}+p^{2}$. Помножимо ліву і праву частину рівняння на $\hat{p}$. Отримаємо $IndexOf\left(p^{sum}\right)*\hat{p}=\left(IndexOf\left(p^{1}\right)+IndexOf\left(p^{2}\right)\right)*\hat{p}$. З доведення наслідку 2 $\left(IndexOf\left(p^{1}\right)+IndexOf\left(p^{2}\right)\right)*\hat{p}=IndexOf\left(p^{1}\right)*\hat{p}+IndexOf\left(p^{2}\right)*\hat{p}$. Крім того з цього ж наслідку $IndexOf\left(p^{sum}\right)*\hat{p}=p^{sum}$, $IndexOf\left(p^{1}\right)*\hat{p}=p^{1}$, $IndexOf\left(p^{2}\right)*\hat{p}=p^{2}$. В результаті отримали, що $p^{sum}=p^{1}+p^{2}$.

Аналізуючи алгебраїчні операції з попереднього пункту можна зробити висновок, що між операціями додавання та множення на множині всіх позиційних представлень перестановок *n*-го порядку та операціями суми та добутку по модулю *n*! існує ізоморфне відображення, яке описується при доведенні наслідків 2, 3 та теореми 4, тобто, якщо p^{1},p^{2} – довільні позиційні представлення перестановок, тоді $IndexOf\left(p^{1}+p^{2}\right)=IndexOf\left(p^{1}\right)+IndexOf\left(p^{2}\right)$ та $IndexOf\left(p^{1}*p^{2}\right)=IndexOf\left(p^{1}\right)*IndexOf\left(p^{2}\right)$.

Висновки. Отримані в роботі результати дозволяють по новому розглянути вже відомі та будувати нові алгоритми для розв’язання задач в яких явно чи неявно використовуються перестановки. В таких алгоритмах певний напрямлений перегляд перестановок можна замінити напрямленим переглядом деякої впорядкованої множини цілих чисел кожне значення з яких є індексом певної перестановки у множині перестановок впорядкованих за базовою перестановкою *Ord*. Крім того, використання прямих лексикографічних обмежень дозволяє будувати різноманітні розбиття впорядкованої множини перестановок, що дає можливість побудови різних схем паралельних алгоритмів. Якщо в розбитті всі підмножини відповідають лексикографічним інтервалам однакової довжини з лексикографічно впорядкованої множини позиційних представлень, тоді є можливість організувати напрямлений перебір не окремих перестановок а цілих підмножин, причому робити це можна за їх індексами.

С.В. Чупов

ЛЕКСИКОГРАФИЧЕСКИ УПОРЯДОЧЕННЫЕ ПЕРЕСТАНОВКИ

Представлен альтернативный способ записи перестановок, который назван позиционным представлением перестановки. На множестве лексикографически упорядоченных позиционных представлений перестановок формулируются разнообразные алгебраические операции над перестановками в их позиционном представлении. Доказывается, что между операциями сложения и умножения на множестве всех позиционных представлений перестановок *n*-го порядка и операциями суммы и произведения по модулю *n*! существует изоморфное отображение.

S.V. Chupov

LEXICOGRAPHICALLY ORDERED PERMUTATIONS

This paper presents an alternative way of writing permutations, which is called the positional representation of permutation. On the set of lexicographically ordered positional representations of permutations various algebraic operations on permutations in their positional representation are formulated. It is proved that there exists an isomorphic mapping between the operations of addition and multiplication on the set of positional representations of permutations of n-th order and the operations of sum and product modulo n!.

1. *Кнут Д.* Искусство программирования, том 3. *Сортировка и поиск*. М.: Вильямс, 2007. С. 824.
2. *Андерсон Дж. Дискретная математика и комбинаторика*. – М.: Вильямс, 2006. С. 960.
3. Стенли Р. Перечислительная комбинаторика. М.: Мир, 1990. С. 440.
4. *Чупов С.В.* Новые подходы к решению задач дискретного программирования на основе лексикографического поиска. *Кибернетика и системный анализ*. 2016. № 4. С. 43 – 54.

Одержано 17.10.2016

Про автора:

Чупов Сергій Вікторович,
кандидат фізико-математичних наук, доцент кафедри системного аналізу
і теорії оптимізації Ужгородського національного університету.
E-mail: sergey.chupov(at)gmail.com