



Data Structures and Algorithms

Алгоритмы.
Генерация перестановок с помощью
циклических сдвигов



Сведение о алгоритме

Сложность по времени в наихудшем случае $O(n!)$

Затраты памяти $O(n)$

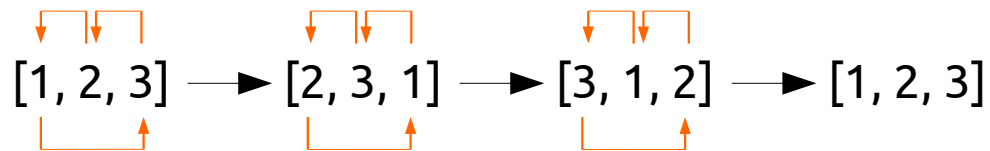


Описание алгоритма

- 1) Создать последовательность размером n . Заполнить ее значениями равными индексам элементов. Создать вспомогательную переменную (в примере k). Установить ее значение равное индексу последнего элемента. Перейти к пункту **2**.
- 2) Вывести на экран последовательность. Установить значение k равное индексу последнего элемента. Перейти к пункту **3**.
- 3) Выполнить циклический сдвиг влево элементов последовательности от первого до k -го. Если k -й элемент последовательности не равен k то перейти к пункту **2**. В противном случае перейти к пункту **4**.
- 4) Указать новое значение $k = k - 1$. Если k равен индексу первого элемента последовательности то **закончить алгоритм**. В противном случае перейти к пункту **3**.



Циклический сдвиг влево



Под циклическим сдвигом влево подразумевается смещение всех значений последовательности на одну позицию влево. При этом первый элемент последовательности становится последним элементом. Если выполнить сдвиг влево определенное количество раз (размер последовательности) то получим опять начальную последовательность. Поэтому сдвиг и называют циклическим.

На рисунке продемонстрирована последовательность циклических сдвигов для последовательности размером 3.



Реализация алгоритма на Python



Python

Реализация алгоритма на Python

Функция циклического сдвига влево первых k элементов списка

```
def left_shift(sequence, k):  
    temp = sequence[0]  
    for i in range(k):  
        sequence[i] = sequence[i+1]  
    sequence[k] = temp
```



Реализация алгоритма на Python

Функция для вывода на экран всех перестановок

```
def print_all_permutation(sequence):  
    k = len(sequence)-1  
    n = k  
    print(sequence)  
    while k > 0:  
        left_shift(sequence, k)  
        if sequence[k] != k:  
            print(sequence)  
            k = n  
        else:  
            k = k-1
```




Java

Реализация алгоритма на Java



Реализация алгоритма на Java

Метод циклического сдвига влево первых k элементов массива

```
public static void leftShift(int[] array, int k) {  
    int temp = array[0];  
    for (int i = 0; i < k; i++) {  
        array[i] = array[i + 1];  
    }  
    array[k] = temp;  
}
```



Реализация алгоритма на Java

Функция для вывода на экран всех перестановок

```
public static void printAllPermutation(int[] array) {  
    int k = array.length - 1;  
    int n = k;  
    System.out.println(Arrays.toString(array));  
    for (; k > 0;) {  
        leftShift(array, k);  
        if (array[k] != k) {  
            System.out.println(Arrays.toString(array));  
            k = n;  
        } else {  
            k = k - 1;  
        }  
    }  
}
```



Список литературы

- 1) Дональд Кнут. «Искусство программирования, том 4, выпуск 2. Генерация всех кортежей и перестановок.» : Пер. с англ. - М. : ООО "И.Д. Вильямс", 2008. - 160 с. ISBN 978-5-8459-1164-3. [71 - 71]