



Data Structures and Algorithms

Генерация перестановок. Алгоритм Хипа.



Сведение о алгоритме

Сложность по времени $O(n!)$



Описание алгоритма

Для первого вызова функции используем последовательность (из элементов которой хотим получить перестановки) и значение k равное ее длине.

- 1) Проверить значение k . В случае если $k == 1$. Вернуть последовательность как текущую перестановку. **Закончить вызов функции.**
- 2) Используя цикл от индекса первого элемента последовательности до k с шагом в 1 выполнить действия:
 - Рекурсивно вызвать функцию передав последовательность и значение $k-1$
 - Если k четное выполнить обмен i и k элемента местами, в противном случае выполнить обмен первого элемента последовательности и k .



Реализация алгоритма на Python



Функция для генерации перестановок

```
def heap_permutation(sequence,k):  
    if k==1:  
        print(sequence)  
        return  
    else:  
        for i in range(k):  
            heap_permutation(sequence,k-1)  
            if k%2 == 0:  
                sequence[i],sequence[k-1]=sequence[k-1],sequence[i]  
            else:  
                sequence[0],sequence[k-1]=sequence[k-1],sequence[0]
```



Java

Реализация алгоритма на Java



Метод для генерации перестановок

```
public static void pernuatatrion(int[] array, int k) {  
    if (k == 1) {  
        System.out.println(Arrays.toString(array));  
        return;  
    }  
    for (int i = 0; i < k; i++) {  
        pernuatatrion(array, k - 1);  
        if (k % 2 == 0) {  
            int temp = array[i];  
            array[i] = array[k - 1];  
            array[k - 1] = temp;  
        } else {  
            int temp = array[0];  
            array[0] = array[k - 1];  
            array[k - 1] = temp;  
        }  
    }  
}
```



Fortran

Реализация алгоритма на Fortran

Процедура для генерации перестановок

```
recursive subroutine permutation(array,k)
  integer, dimension(:), intent(inout) :: array
  integer, intent(in) :: k
  integer :: i, temp
  if (k==1) then
    write(*,*) array
    return
  end if
  do i = 1,k
    call permutation(array,k-1)
    if(mod(k,2)==0) then
      temp = array(i)
      array(i) = array(k)
      array(k) = temp
    else
      temp = array(1)
      array(1) = array(k)
      array(k) = temp
    end if
  end do
end subroutine permutation
```



Список литературы

1) Sedgewick, Robert. "a talk on Permutation Generation Algorithms"