

**МИНОБРНАУКИ РОССИИ**  
**ФГБОУ ВО «СГУ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

**АЛГОРИТМ ШЕЙКЕРНОЙ СОРТИРОВКИ**  
**ЛАБОРАТОРНАЯ РАБОТА**

студента 3 курса 331 группы  
специальности 100501 — Компьютерная безопасность  
факультета КНиИТ  
Окунькова Сергея Викторовича

Проверил  
доцент

\_\_\_\_\_

А. Н. Гамова

## СОДЕРЖАНИЕ

1	Описание алгоритма .....	3
2	Эффективность алгоритма.....	4
3	Реализация .....	5
4	Тестирование программы .....	6
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	7

## 1 Описание алгоритма

Шейкерная сортировка (Cocktail sort), она же сортировка перемешиванием, она же двунаправленная сортировка — по сути всего лишь оптимизированный алгоритм пузырьковой сортировки, в основе которой также лежит сравнение двух соседних элементов. Единственное отличие состоит лишь в том, что теперь это происходит в двух направлениях поочередно, постепенно сужая диапазон сортировки. В итоге за один проход в конец массива “всплывает” максимальный элемент из диапазона, а за следующий проход — в начало массива минимальный. Эти элементы можно больше не рассматривать и таким образом диапазон сужается с двух сторон.

## 2 Эффективность алгоритма

Сложность работы алгоритма в худшем случае определяется как  $O(n^2)$ , так как суммарное количество сравнений равно  $(n - 1)\frac{n}{2}$ , а число обменов равно  $(n - 1)\frac{n}{2}$ , где  $n$  - это размер входного массива, что делает эту сортировку не самой эффективной в классе сортировок.

Сложность работы алгоритма в лучшем случае определяется как  $O(n)$ , так как суммарное количество сравнений равно  $(n - 1)\frac{n}{2}$ , а число обменов равно 0, где  $n$  - это размер входного массива. Этот случай достигается тогда, когда на вход подается уже отсортированный массив.

Главным плюсом данной сортировки является то, что она не требует выделения дополнительной памяти.

### 3 Реализация

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

//ф-ция "шейкер"-сортировки
void myShakerSort(vector<int> &arr, const int size) {
    int leftMark = 1;
    int rightMark = size - 1;
    while (leftMark <= rightMark) {
        for (int i = rightMark; i >= leftMark; i--)
            if (arr[i - 1] > arr[i]) swap(arr[i], arr[i-1]);
        leftMark++;

        for (int i = leftMark; i <= rightMark; i++)
            if (arr[i - 1] > arr[i]) swap(arr[i], arr[i-1]);
        rightMark--;
    }
}

int main(void) {
    setlocale(LC_ALL, "rus");

    int size;
    cout << "Размер массива: ";
    cin >> size;
    vector<int> arr (size);
    for (int i = 0; i < size; ++i) {
        cin >> arr[i];
    }

    myShakerSort(arr, size); // сортировка

    cout << "Массив после сортировки:\n";
    for (int i = 0; i < size; ++i) {
        cout << arr[i] << " ";
    }
    cout << endl;
}
```

#### 4 Тестирование программы

```
sokunkov@C11310 complexity_calculation % g++ -std=c++11 lab1.cpp
sokunkov@C11310 complexity_calculation % ./a.out
Размер массива: 10
-1 3 2 0 9 7 -8 10 110 -100
Массив после сортировки:
-100 -8 -1 0 2 3 7 9 10 110
sokunkov@C11310 complexity_calculation %
```

Рисунок 1 – Тест1

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

- 1 Скиена Стивен "Алгоритмы. Руководство по разработке 2018 год. Яз. рус.
- 2 Нииколаус Вирт "Алгоритмы и структуры данных 2008 год. Яз. рус.