

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №4**  
**по дисциплине «Алгоритмы и структуры данных»**  
**Тема: Сортировки**

Студент гр. 9304

Алексееenko Б.

Преподаватель

Филатов Ар.Ю.

Санкт-Петербург

2020

## **Цель работы.**

Ознакомиться с методами сортировки, реализовать с помощью языка программирования C++ один из методов сортировки для любых типов данных.

## **Задание.**

Вариант 5.

Гибрид сортировки пузырьком и сортировки выбором; шейкерная сортировка.

## **Описание алгоритма.**

Шейкерная сортировка — модификация пузырьковой сортировки. Производится многократный прогон по массиву, соседние элементы сравниваются и, в случае необходимости, меняются местами. При достижении конца массива направление меняется на противоположное. Таким образом по очереди выталкиваются крупные и мелкие элементы массива в конец и начало структуры соответственно. Коктейльная сортировка ещё называется двухсторонней сортировкой простыми обменами. Есть аналогичная модификация и для сортировки выбором. Асимптотика у алгоритма такая же, как и у сортировки пузырьком.

Достоинства:

Шейкерная сортировка является устойчивой.

Решает недостаток пузырьковой сортировки, связанный с «легкими» числами в конце массива или, как их еще называют, черепахами.

Реальное время работы выше, чем у сортировки пузырьком.

Недостатки:

В худшем случае скорость работы остается такой же как и у сортировки пузырьком.

## **Выполнение работы.**

Работает алгоритм в коде следующим образом: мы поддерживаем два указателя — begin и end, которые показывают, какая часть массива еще не была

отсортирована. Как и в сортировке пузырьком мы идем по массиву и попутно сравниваем элементы, но при достижении конца массива, мы начинаем идти в обратную сторону, пока не будут отсортированы все элементы.

### **Выводы.**

Было проведено ознакомление с методами сортировки, был реализован метод шейкерной сортировки с помощью языка программирования C++.

Была разработана программа, которая сортирует массив с помощью шейкерной сортировки. Хотя и сложность этого алгоритма такая же как и у сортировки пузырьком (в лучшем случае -  $O(n)$ , в худшем —  $O(n^2)$ ), но на деле реальное время работы, на мой взгляд будет лучше.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

void shakersort(int* left, int* right) {
    int sz = right - left;
    if (sz <= 1) return;
    bool b = true;
    int* beg = left - 1;
    int* end = right - 1;
    while (b) {
        b = false;
        beg++;
        for (int* i = beg; i < end; i++) {
            if (*i > *(i + 1)) {
                cout << "swap: " << *i << " & " << *(i + 1) << endl;
                swap(*i, *(i + 1));
                b = true;
            }
        }
        if (!b) break;
        end--;
        cout << "Move another way" << endl;
        for (int* i = end; i > beg; i--) {
            if (*i < *(i - 1)) {
                cout << "swap: " << *i << " & " << *(i - 1) << endl;
                swap(*i, *(i - 1));
                b = true;
            }
        }
    }
}

int main() {
    int* data;
    string name_;
    ifstream f;
    cout << "Enter a file name.\n";
    cin >> name_;
    f.open(name_);
    if (!f.is_open()) {
        cout << "Invalid file name.\n";
        return 1;
    }
    while (!f.eof()) {
        int size;
        f >> size;
        data = new int[size];

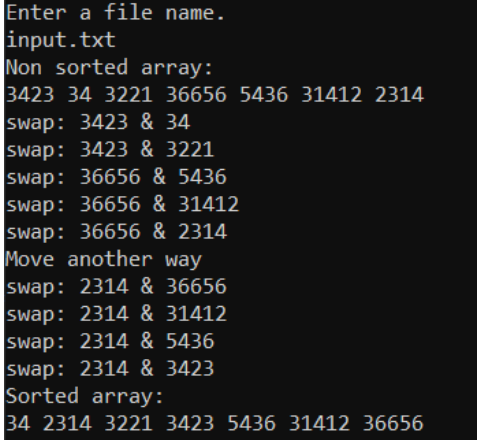
        for (int i = 0; i < size; i++) {
            f >> data[i];
        }
        cout << "Non sorted array: " << endl;
        for (int i = 0; i < size; i++)
            cout << data[i] << " ";
        cout << endl;
        shakersort(&data[0], &data[size]);
    }
}
```

```
        cout << "Sorted array:\n";
        cout << endl;
        for (int i = 0; i < size; i++)
            cout << data[i] << " ";
        delete[] data;
    }
    return 0;
}
```

## ПРИЛОЖЕНИЕ Б

### ТЕСТИРОВАНИЕ

Таблица 1 — Примеры тестовых случаев.

№ п/п	Входные данные	Выходные данные	Комментарии
1.	17 3423 34 3221 36656 5436 31412 2314	 <pre> Enter a file name. input.txt Non sorted array: 3423 34 3221 36656 5436 31412 2314 swap: 3423 &amp; 34 swap: 3423 &amp; 3221 swap: 36656 &amp; 5436 swap: 36656 &amp; 31412 swap: 36656 &amp; 2314 Move another way swap: 2314 &amp; 36656 swap: 2314 &amp; 31412 swap: 2314 &amp; 5436 swap: 2314 &amp; 3423 Sorted array: 34 2314 3221 3423 5436 31412 36656           </pre> <p>Рисунок 1 — Пример работы программы.</p>	
2.	20 123 34 344 324 32 5432 5452324 432 324 3444 4324 3211 90 786 1234 3432 1111 12 432 91	<pre> Enter a file name. input.txt Non sorted array: 123 34 344 324 32 5432 5452324 432 324 3444 4324 3211 90 786 1234 3432 1111 12 432 91  swap: 123 &amp; 34 swap: 344 &amp; 324 swap: 344 &amp; 32 swap: 5452324 &amp; 432 swap: 5452324 &amp; 324 swap: 5452324 &amp; 3444 swap: 5452324 &amp; 4324 swap: 5452324 &amp; 3211           </pre>	

		swap: 5452324 & 90 swap: 5452324 & 786 swap: 5452324 & 1234 swap: 5452324 & 3432 swap: 5452324 & 1111 swap: 5452324 & 12 swap: 5452324 & 432 swap: 5452324 & 91 Move another way swap: 91 & 5452324 swap: 12 & 91 swap: 12 & 1111 swap: 12 & 3432 swap: 12 & 1234 swap: 12 & 786 swap: 12 & 90 swap: 12 & 3211 swap: 12 & 4324 swap: 12 & 3444 swap: 12 & 324 swap: 12 & 432 swap: 12 & 5432 swap: 12 & 344 swap: 12 & 32 swap: 12 & 324 swap: 12 & 123 swap: 324 & 32 swap: 5432 & 432 swap: 5432 & 324	
--	--	--	--

		swap: 5432 & 3444 swap: 5432 & 4324 swap: 5432 & 3211 swap: 5432 & 90 swap: 5432 & 786 swap: 5432 & 1234 swap: 5432 & 3432 swap: 5432 & 1111 swap: 5432 & 91 swap: 5432 & 432 Move another way swap: 91 & 432 swap: 91 & 1111 swap: 91 & 3432 swap: 91 & 1234 swap: 90 & 91 swap: 90 & 3211 swap: 90 & 4324 swap: 90 & 3444 swap: 90 & 324 swap: 90 & 432 swap: 90 & 344 swap: 32 & 90 swap: 32 & 123 swap: 123 & 90 swap: 432 & 324 swap: 4324 & 3211 swap: 4324 & 91 swap: 4324 & 786	
--	--	---	--



		swap: 4324 & 1234 swap: 4324 & 3432 swap: 4324 & 1111 swap: 4324 & 432 Move another way swap: 432 & 4324 swap: 432 & 1111 swap: 432 & 3432 swap: 432 & 1234 swap: 91 & 432 swap: 91 & 3211 swap: 91 & 3444 swap: 91 & 432 swap: 91 & 324 swap: 91 & 344 swap: 91 & 324 swap: 344 & 324 swap: 3444 & 3211 swap: 3444 & 432 swap: 3444 & 786 swap: 3444 & 1234 swap: 3444 & 3432 swap: 3444 & 1111 Move another way swap: 1111 & 3444 swap: 1111 & 3432 swap: 432 & 786 swap: 3211 & 786 swap: 3211 & 1111	
--	--	--	--

		swap: 3211 & 1234 Move another way Sorted array: 12 32 34 90 91 123 324 324 344 432 432 786 1111 1234 3211 3432 3444 4324 5432 5452324	
3.	5 4 776 12 34 5	Enter a file name. input.txt Non sorted array: 4 776 12 34 5 swap: 776 & 12 swap: 776 & 34 swap: 776 & 5 Move another way swap: 5 & 776 swap: 5 & 34 Sorted array: 4 5 12 34 776	