**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
 РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет информационных технологий**

**Кафедра параллельных вычислений**

**ОТЧЕТ**

**О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

«ОПРЕДЕЛЕНИЕ ВРЕМЕНИ РАБОТЫ ПРИКЛАДНЫХ ПРОГРАММ»

студента 2 курса, группы 22210

**Бирюля Михаила Сергеевича**

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

М. А. Мичуров

Новосибирск 2023

**СОДЕРЖАНИЕ**

[**ЦЕЛЬ РАБОТЫ 3**](#_gjdgxs)

[**ЗАДАЧИ 3**](#_ep6k9sqf4iso)

[**ОПИСАНИЕ РАБОТЫ 4-6**](#_1fob9te)

[**ЗАКЛЮЧЕНИЕ 7**](#_78k7oto7vlq1)

# ЦЕЛЬ РАБОТЫ

Изучение и использование методик измерения времени работы программы.

**ЗАДАЧИ**

1. Написать программу на языке C или C++, которая реализует выбранный алгоритм из задания.

2. Проверить правильность работы программы на нескольких тестовых наборах входных данных.

3. Выбрать значение параметра N таким, чтобы время работы программы было порядка 15 секунд.

4. Составить отчет по лабораторной работе.

**ОПИСАНИЕ РАБОТЫ:**

**Ссылка на репозиторий с кодом:** [termoler/evm\_labs (github.com)](https://github.com/termoler/evm_labs)

**1) Описание алгоритма:**

В роли испытуемой программы я выбрал алгоритм сортировки Bubble Sort. Данная сортировка является одной из самых простейших в реализации, так как в ее основе лежит легкое для понимания попарное сравнивание элементов входного массива. То есть, за каждый проход элементы последовательно сравниваются попарно и, если порядок в паре неверный, выполняется перестановка элементов. Проходы по массиву повторяются �−1N - 1 раз или до тех пор, пока на очередном проходе не окажется, что обмены больше не нужны, что означает — массив отсортирован.

При каждом проходе алгоритма по внутреннему циклу очередной наибольший элемент массива ставится на своё место в конце массива рядом с предыдущим «наибольшим элементом», а наименьший элемент перемещается на одну позицию к началу массива («всплывает» до нужной позиции, как пузырёк в воде — отсюда и название алгоритма).

**2) Листинг:**

Для реализации алгоритма, я решил использовать язык программирования С++. Таким образом код имеет следующий вид:

#include <iostream>  
#include <cstdlib>  
#include <string>  
  
class Array{  
 int\* arr;  
 size\_t size;  
public:  
 explicit Array(size\_t size): arr(new int[size]{}), size(size){  
 for(size\_t i = 0; i < size; i++){  
 arr[i] = rand();  
// std::cin >> arr[i];  
 }  
 }  
 int &operator[](size\_t i){  
 return arr[i];  
 }  
 size\_t get\_size() const {  
 return size;  
 }  
 ~Array(){  
 delete[] arr;  
 }  
};  
  
void swap(int\* x, int\* y){  
 int tmp = \*x;  
 \*x = \*y;  
 \*y = tmp;  
}  
  
void bubble\_sort(Array& arr){  
 for (int i = 0; i < arr.get\_size(); i++) {  
 for (int j = 0; j < arr.get\_size(); j++) {  
 if (arr[j] > arr[j + 1]) {  
 swap(&arr[j], &arr[j + 1]);  
 }  
 }  
 }  
}  
  
void print(Array& arr) {  
 size\_t size = arr.get\_size();  
 for (size\_t i = 0; i < size; i++) {  
 std::cout << arr[i] << ' ';  
 }  
}  
  
int main(int argv, char\*\* argc){  
 if(argv == 2){  
 size\_t size = std::stoi(argc[1]);  
 Array arr(size);  
 bubble\_sort(arr);  
 }  
// print(arr);  
 return 0;  
}

Для автоматизации получения результатов измерения времени, было решено написать Bash скрипт следующего вида:

#!/bin/bash

echo "Введите размер массива: "

read n

for i in $(seq 7); do time ./untitled15.exe $n

done

**3) Результаты:**

После проверки правильности работы программы, экспериментальным путем было выявлено что времени исполнения работы программы в примерно 15 секунд соответствует значение параметра равное 40000.

Далее представлены результаты выполнения в виде таблицы и соответствующего графика.

|  |  |
| --- | --- |
| № | Результат измерения (сек.) |
| 1: | 15,571 |
| 2: | 17,474 |
| 3: | 15,579 |
| 4: | 17,172 |
| 5: | 15,606 |
| 6: | 15,564 |
| 7: | 15,566 |
| 8: | 16,132 |
| 9: | 15,478 |
| 10: | 15,593 |

# 

# ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы мне удалось успешно реализовать алгоритм сортировки “Bubble Sort” для массива размера N. Я изучил различные методы измерения времени выполнения программы и, используя утилиту “time”, научился определять продолжительность работы моей программы.