**Московский государственный технический**

**Университет им. Н.Э. Баумана**

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №6 (сдвоенная пятая)

«Работа с файлами, сортировка и измерение времени выполнения программы на языке программирования Rust»

Выполнил:

студент группы ИУ5Ц-52Б

Мефодьев Илья

Проверил:

Нардид А.Н.

2024 г.

**Задание**

Разработать программу для получения массива чисел из файла и его сортировки.

1. Прочитать текстовый файл и поместить содержимое в целочисленный массив. Текстовый файл содержит случайное число строк, в каждой из которых записано целое число. Числа могут повторяться и быть отрицательными.

2. Отсортировать полученный массив оптимальным способом

3. Записать отсортированный массив в другой файл.

4. Измерить время выполнения программы, время сортировки.

**Текст программы**

*// взял вторую лабу и выполню то же задание на Rust.*

use std::fs;

use std::fs::File;

use std::io::prelude::\*;

use std::time::{Duration, Instant};

fn quick\_sort<T: Ord>(*mut* arr: Vec<T>) -> Vec<T> { *// шаблонная функция, работает с любыми сравниваемыми типамиЫ*

    if arr.len() <= 1 {

        return arr;

    }

    let pivot = arr.remove(0);

    let *mut* left = vec![];

    let *mut* right = vec![];

    for item in arr {

        if item <= pivot {

            left.push(item);

        } else {

            right.push(item);

        }

    }

    let *mut* sorted\_left = quick\_sort(left);

    let *mut* sorted\_right = quick\_sort(right);

    sorted\_left.push(pivot);

    sorted\_left.append(&*mut* sorted\_right);

    sorted\_left

}

fn main() {

    let start = Instant::now();

    let file\_path = "rand.txt";

    let contents = fs::read\_to\_string(file\_path)

        .expect("Не можем прочитать файл!");

*//println!("With text:\n{contents}");*

    let nums  = contents.lines();

    let *mut* count = 0;

*// for num in nums{*

*//     //println!("{}", num);*

*//     count += 1;*

*// }*

    let *mut* numsVec: Vec<i32> = Vec::new();

*//let mut arr: [i32; 9];*

*//let mut i = 0;*

    for num in nums{

        let el = num.parse::<i32>().unwrap();

        numsVec.push(el);

*//println!("{}", el);*

        count+=1;

*//arr[i] = el;*

*//i+=1;*

    }

*//println!("completed creating Vector!");*

    let start\_sort = Instant::now();

    let res = quick\_sort(numsVec);

    let dur\_sort = start\_sort.elapsed();

*//println!("completed sorting Vector!");*

    let *mut* result :String = "".to\_string();

*//for el in numsVec{*

    for el in res{

*//result += el.to\_string();*

        result.push\_str(&el.to\_string());

        result.push('\n');

    }

*//println!("{result}");*

    let *mut* file = fs::File::create("SortedRust.txt").expect("Ошибка чтения файла!");

*//file.write\_all(result.as\_bytes());*

    write!(file, "{result}").expect("Ошибка записи файла!");

    let end = start.elapsed();

    println!("Массив отсортирован. Время сортировки: {:?}ms;", dur\_sort);

    println!("Время работы всей программы: {:?}ms;", end);

}

**Скриншоты с примерами выполнения программы**

