**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 №2

«Работа с файлами, сортировка и измерение времени выполнения программы на языке программирования JavaScript (Node JS)»

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**

const start = new Date().getTime();

const fs = require("fs");

let str = fs.readFileSync("rand.txt", 'utf-8'); *// читаем файл синхронно, иначе логические ошибки*

let arrStrs = str.split("\n");

let arrInts = [];

for(let i = 0; i < arrStrs.length; i++){

    arrInts[i] = Number(arrStrs[i]);

}

function getQuickSortedArr(arr){ *// OK*

    if (arr.length <= 1) return arr;

    let opora = arr[arr.length - 1];

    const left = [];

    const right = [];

   for (let i = 0; i < arr.length - 1; i++) {

      if (opora >= arr[i]) {

        left.push(arr[i]);

      } else {

        right.push(arr[i]);

      }

    }

    return [...getQuickSortedArr(left), opora, ...getQuickSortedArr(right)];

*// QuickSort - возвращает массив, но т.к. тут рекурсия доходит до массива с длинной 1,*

*// надо в вышестоящих функциях достать эти элементы и запихнуть в новый массив.*

*// для этого юзаем оператор расширения ...*

*// (иначе получим массив массивов)*

}

function getBubbleSortedArr(arr){ *// ok but too slow*

    let res = arr;

    for (let i = 0; i < res.length; i++) {

        for (let j = 0; j < res.length - i; j++) {

            if(res[j+1] < res[j]){

                let t = res[j];

                res[j] = res[j+1];

                res[j+1] = t;

            }

        }

    }

    return res;

}

const startSort = new Date().getTime(); *// отдельно посчитаем и время непосредственно сортировки*

*//let res = getBubbleSortedArr(arrInts); // 26 секунд сортировка...*

let res = getQuickSortedArr(arrInts); *// 60ms*

const endSort = new Date().getTime();

let strToFile = "";

for (let i = 0; i < res.length; i++) {

    strToFile += res[i].toString();

    if(i!=res.length - 1){strToFile += "\n";}

}

fs.writeFile("SortedJS.txt", strToFile, function(error){

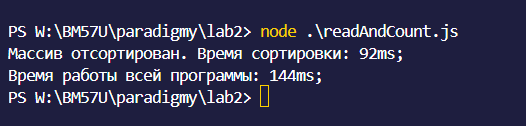
    if(error) {return console.log(error)};

});

const end = new Date().getTime();

console.log(`Массив отсортирован. Время сортировки: ${endSort - startSort}ms;\nВремя работы всей программы: ${end - start}ms;`);

**Скриншоты с примерами выполнения программы**



Прочесть файл SortedJS.txt, чтобы убедиться в результате.