

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**"МИРЭА – Российский технологический университет"**

# РТУ МИРЭА

Институт информационных технологий (ИТ)

Кафедра МОСИТ

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

Исследование сортировок 2

**«АЛГОРИТМЫ ОБРАБОТКИ ДАННЫХ»**

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил студент группы ИКБО-16-19 . | |  | Павлов С.П. |
| Принял  д.т.н. , проф |  |  | Кораблин Ю.П. |
| Лабораторные работы выполнены | «\_\_» \_\_\_\_\_\_\_ 2020 г. |  |
|  |  |  |  |
| «Зачтено» | «\_\_» \_\_\_\_\_\_\_ 2020 г.  Москва 2020 |  |  |
|

**CОДЕРЖАНИЕ**

|  |  |  |
| --- | --- | --- |
| 1. | Программа сортировки реализованная на с++ | 3 |
| 2. | Результаты выполнения программы | 4 |
|  |
|  |  |  |
|  |  |  |
|  |  |  |

**# Сортировка естественным слиянием**

vector<int> mergeArrays(vector<int>& arr1, vector<int>& arr2)

{

int i = 0, j = 0, k = 0;

vector<int> arr3(arr1.size() + arr2.size());

while (i < arr1.size() && j < arr2.size())

{

if (arr1[i] < arr2[j])

arr3[k++] = arr1[i++];

else

arr3[k++] = arr2[j++];

}

while (i < arr1.size())

arr3[k++] = arr1[i++];

while (j < arr2.size())

arr3[k++] = arr2[j++];

sort(arr3.begin(), arr3.end());

return arr3;

}

vector<int> Sorts::natural\_merge\_sort() {

this->start\_clock = clock();

vector<int> \_temp = this->input;

vector<vector<int>> \_mas\_of\_mases;

int \_counter = 0;

if (\_temp.size() >= 10)

for (size\_t i = 0; i < \_temp.size(); i += \_temp.size() / 10) {

\_mas\_of\_mases.push\_back(vector<int>());

for (size\_t j = 0; j < \_temp.size() / 10; ++j)

\_mas\_of\_mases[\_counter].push\_back(\_temp[j]);

\_counter++;

}

else

for (size\_t i = 0; i < \_temp.size(); i++) {

\_mas\_of\_mases.push\_back(vector<int>());

\_mas\_of\_mases[\_counter].push\_back(\_temp[i]);

\_counter++;

}

while (\_mas\_of\_mases.size() != 1) {

for (size\_t i = 0; i < \_mas\_of\_mases.size() - 1; i += 2) {

\_mas\_of\_mases[i] = mergeArrays(\_mas\_of\_mases[i], \_mas\_of\_mases[i + 1]);

\_mas\_of\_mases[i + 1].clear();

}

for (size\_t i = 0; i < \_mas\_of\_mases.size(); ++i)

if (\_mas\_of\_mases[i].size() == 0)

\_mas\_of\_mases.erase(\_mas\_of\_mases.begin() + i);

}

this->estimated\_time = (clock() - start\_clock) / 1000.0;

return \_mas\_of\_mases[0];

}

Результаты выполнения программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sort | 1000 | 10000 | 100000 | 1000000 |
| Natural merge | 0.0001s | 0.001s | 0.008s | 0.083s |