|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
|  | | |
|  | | |
|  | | |
| Практическое задание № 3 | | |
| по дисциплине «Методы построения и анализа алгоритмов» | | |
|  | | |
| **кафедра теоретической и прикладной информатики** | | |
|  | | |
|  | Факультет: | ПМИ |
| Группа: | ПМИ-03 |
| Бригада: | Место для ввода текста. |
| Студенты: | Сидоров Даниил, |
|  | Малыгин Сергей |
|  |  |
| Преподаватели: | Щукин Георгий Анатольевич |
|  |  |
|  | | |
| Новосибирск | | |
| 2021 | | |

**1.Результаты замеров:**

|  |  |  |  |
| --- | --- | --- | --- |
| N | Count sort(vector) | Count sort(map) | Quick sort |
| 100 000 | 0.0016809 sec | 0.0481835 sec | 0.0109699 sec |
| 1 000 000 | 0.0180214 sec | 0.572942 sec | 0.112269 sec |
| 10 000 000 | 0.17427 sec | 5.49445 sec | 0.950683 sec |
| 100 000 000 | 1.963 sec | 27.4952 sec | 9.35932 sec |

**2.Программа:**

**Main.cpp**

#include "count\_sort.h"

#include <vector>

#include <iostream>

#include <algorithm>

#include <ctime>

using namespace std;

#define CATCH\_CONFIG\_RUNNER

#include "catch.hpp"

int min\_find(const vector<int> &array)

{

int result(0);

for (unsigned int i(0); i < array.size(); i++)

if (result > array[i])

result = array[i];

return result;

}

int max\_find(const vector<int>& array)

{

int result(0);

for (unsigned int i(0); i < array.size(); i++)

if (result < array[i])

result = array[i];

return result;

}

const unsigned int MIN = 100000;

const unsigned int MAX = 1000000000;

int main(int argc, char \*\*argv)

{

int result = Catch::Session().run(argc, argv);

srand(time(0));

vector<int> Data1;

vector<int> Data2;

vector<int> Result1;

vector<int> Result2;

for (unsigned int i(MIN); i < MAX; i \*= 10)

{

for (int j = 0; j < i; j++)

{

Data1.push\_back(rand() % i);

}

int min1 = min\_find(Data1);

int max1 = max\_find(Data1);

auto t1 = std::chrono::high\_resolution\_clock::now();

count\_sort(Data1, min1, max1);////VECTOR

auto t2 = std::chrono::high\_resolution\_clock::now();

auto seconds = std::chrono::duration<double>(t2 - t1).count();

cout << "N = " << i << endl;

cout << "Count sort(vector) === Time: " << seconds << " sec." << endl;

t1 = std::chrono::high\_resolution\_clock::now();

count\_sort\_map(Data1);////MAP

t2 = std::chrono::high\_resolution\_clock::now();

seconds = std::chrono::duration<double>(t2 - t1).count();

//cout << "N = " << i << endl;

cout << "Count sort(map) === Time: " << seconds << " sec." << endl;

t1 = std::chrono::high\_resolution\_clock::now();

sort(Data1.begin(), Data1.end());////QUICK SORT

t2 = std::chrono::high\_resolution\_clock::now();

seconds = std::chrono::duration<double>(t2 - t1).count();

cout << "Quick sort === Time: " << seconds << " sec." << endl << endl;

Data1 = {};

}

return result;

}

**Count\_sort.cpp**

#include "count\_sort.h"

std::vector<int> count\_sort(const std::vector<int> &array, int min, int max)

{

std::vector<int> result;

std::vector<int> count(max - min + 1);

for (unsigned int i(0); i < array.size(); i++)

count[array[i] - min]++;

for (unsigned int i(0); i < count.size(); i++)

for (unsigned int j(0); j < count[i]; j++)

result.push\_back(i + min);

return result;

}

std::vector<int> count\_sort\_map(const std::vector<int>& array)

{

std::vector<int> result;

std::map<int, unsigned int> count;

for (int i(0); i < array.size(); i++)

count[array[i]]++;

for (std::map<int, unsigned int> ::iterator it = count.begin(); it != count.end(); it++)

for (unsigned int j(0); j < it->second; j++)

result.push\_back(it->first);

return result;

}

**Count\_sort.h**

#pragma once

#include <vector>

#include <map>

std::vector<int> count\_sort(const std::vector<int> &array, int min, int max);

std::vector<int> count\_sort\_map(const std::vector<int>& array);