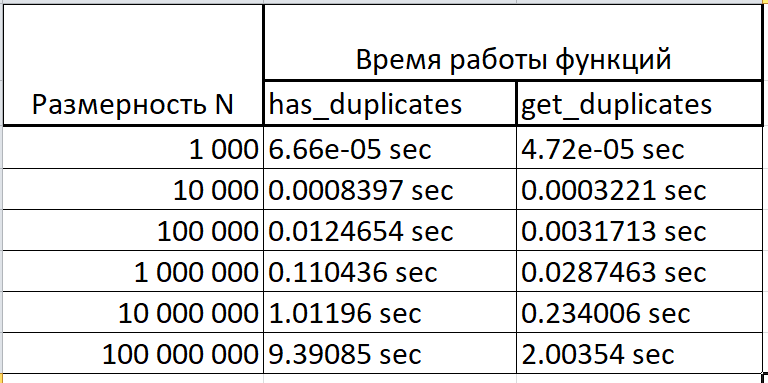
|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
|  | | |
|  | | |
|  | | |
| Практическое задание № 2 | | |
| 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:**

Оптимизация по скорости O2:



**2.Программа:**

#include<iostream>

#include<algorithm>

#include<vector>

#include <chrono>

#include <ctime>

using namespace std;

bool has\_duplicates(vector<int> &Data)

{

int prd;

bool flag = 0;

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

for (int &elem : Data)

{

if (flag == 1)

{

if (elem == prd) return true;

prd = elem;

}

else

{

flag = 1;

prd = elem;

}

}

//cout << "Duplicates not found"<< endl;

return false;

}

void output(vector<int> &result)

{

for (int &elem : result) {

cout << elem << " ";

}

cout << endl;

result = {};

}

vector <int> get\_duplicates(vector<int> &Data, vector<int> &result)

{

int prd;

int prdr;

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

bool flag = 0, flag2 = 0;

for (int &elem : Data)

{

if (flag == 1)

{

if (elem == prd)

{

if (flag2 == 1)

{

if (prdr != elem)

{

prdr = elem;

result.push\_back(elem);

}

else prdr = elem;

}

else

{

flag2 = 1;

result.push\_back(elem);

prdr = elem;

}

}else

{

prd = elem;

}

}

else

{

flag = 1;

prd = elem;

}

}

return result;

}

int main()

{

srand(time(0));

vector<int> Data;

vector<int> result;

Data = {};

if (has\_duplicates(Data))

{

get\_duplicates(Data, result); output(result);

}

Data = { 1 };

if (has\_duplicates(Data))

{

get\_duplicates(Data, result); output(result);

}

Data = { 2, 1, -4, 7 };

if (has\_duplicates(Data))

{

get\_duplicates(Data, result); output(result);

}

Data = { 2, -3, 0, 2, 7, 1 };

if (has\_duplicates(Data))

{

get\_duplicates(Data, result); output(result);

}

Data = { 2, -3, 0, 2, 1, -3, 4, 1, -1, 2 };

if (has\_duplicates(Data))

{

get\_duplicates(Data, result); output(result);

}

Data = { 4, 4, 4, 4 };

if (has\_duplicates(Data))

{

get\_duplicates(Data, result); output(result);

}

Data = {};

for (int i = 1000; i <= 100000000; i = i \* 10)

{

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

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

}

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

if (has\_duplicates(Data))

{

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

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

cout << "N = " << i << ": " << "Time: " << seconds << " sec." << std::endl;

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

get\_duplicates(Data,result);

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

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

cout << "N = " << i << ": " << "Time: " << seconds << " sec." << std::endl<<endl;

result = {};

}

Data = {};

}

cout << endl << endl;

for (int i = 1000; i <= 100000000; i = i \* 10)

{

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

Data.push\_back(j);

}

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

has\_duplicates(Data);

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

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

cout << "N = " << i << ": " << "Time: " << seconds << " sec." << std::endl;

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

get\_duplicates(Data, result);

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

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

cout << "N = " << i << ": " << "Time: " << seconds << " sec." << std::endl << endl;

result = {};

Data = {};

}

}