ОТЧЕТ

по лабораторной работе № 5

«Поиск кратчайших путей на графах»

по дисциплине

«ОСНОВЫ ТЕОРИИ АЛГОРИТМОВ»

Выполнил студент группы ИС/б-22о

Горбенко К.Н.

Проверил:

Карлусов В.Ю.

* 1. ЦЕЛЬ РАБОТЫ

Целью работы является изучение алгоритмов поиска кратчайших путей на графах на примере метода динамического программирования.

* 1. ВАРИАНТ ЗАДАНИЯ

Для варианта № 17 задан граф, изображенный на рисунке № 1:

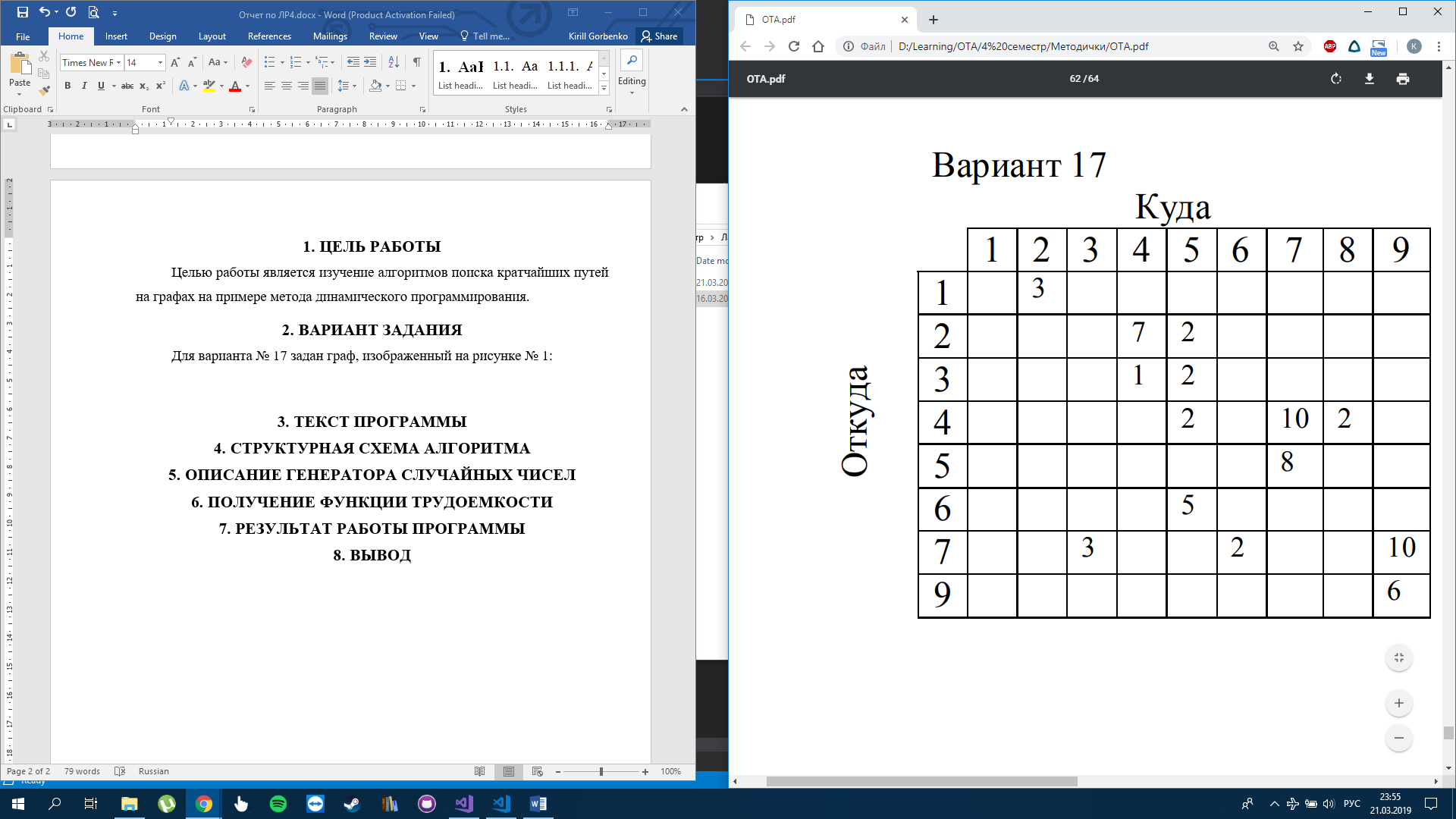


Рис. 1 – Исходный граф для варианта № 17

* 1. СТРУКТУРНЫЕ СХЕМЫ АЛГОРИТМОВ
     1. Структурная схема алгоритма поиска кратчайших путей методом динамического программирования изображена на рисунке № 2:

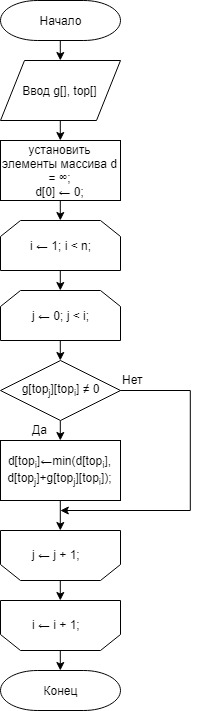


Рис. 2 – Схема алгоритма поиска кратчайших путей в графе методом динамического программирования

* + 1. Структурная схема алгоритма топологической сортировки изображена на рисунке 3, схема подпрограммы dfs, использующейся в алгоритме изображена на рисунке № 4:

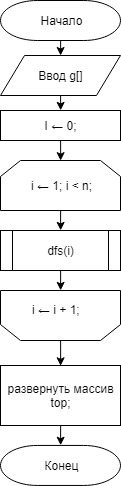


Рис. 3 – Схема основной части алгоритма топологической сортировки

Схема подпрограммы dfs:

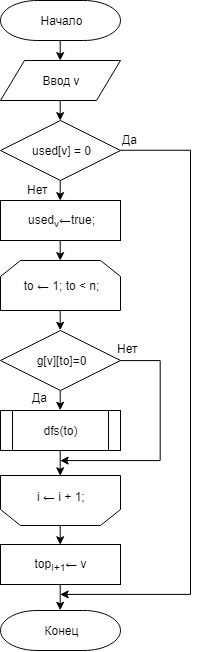


Рис. 4 – Схема подпрограммы dfs, включенной в алгоритм топологической сортировки

* 1. ТЕКСТЫ ПРОГРАММ
     1. Текст программы поиска кратчайшего пути методом динамического программирования:

const int INF = 1000000000;

bool used[9] = { 0 };

int top[9] = { 0 };

int g[9][9] = { 0 };

int n = 9;

int l;

int s = 0;

int f = 8;

int d[9] = { 0 };

int dfs(int v) {

if (used[v])

return 0;

used[v] = true;

for (int to = 0; to < n; to++)

if (g[v][to])

dfs(to);

top[l++] = v;

}

int topSort() {

l = 0;

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

dfs(i);

reverse(top, top + l);

return 0;

}

int solve() {

int i, j;

for (i = 0; i < n; i++)

d[i] = INF;

d[s] = 0;

for (i = 1; i < n; i++)

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

if (g[top[j]][top[i]])

d[top[i]] = min(d[top[i]], d[top[j]] + g[top[j]][top[i]]);

return 0;

}

int main()

{

ifstream myFile("D:\\Learning\\ОТА\\4 семестр\\Лабораторные работы\\ЛР4\\ShortestWayInGraph\\ShortestWayInGraph\\Debug\\file.txt");

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

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

myFile >> g[i][j];

if (g[i][j] == -1) g[i][j] = INF;

}

}

clock\_t startTime = clock();

//topSort();

//solve();

solve\_d();

double timeTaken = double(clock() - startTime) / (double)CLOCKS\_PER\_SEC;

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

cout << d[i] << " ";

}

cout << endl;

cout << "Time taken: " << timeTaken << " seconds." << endl;

}

* + 1. Текст программы поиска кратчайшего пути с использованием алгоритма Дейкстры:

const int INF = 1000000000;

bool used[9] = { 0 };

int top[9] = { 0 };

int g[9][9] = { 0 };

int n = 9;

int l;

int s = 0;

int f = 8;

int d[9] = { 0 };

void solve\_d() {

int i, j,

v,

to,

len;

d[s] = 0;

for (i = 0; i < n; i++) {

v = -1;

for (j = 0; j < n; j++)

if (!used[j] && (v == -1 || d[j] < d[v]))

v = j;

used[v] = true;

for (to = 0; to < n; to++) {

if (g[v][to]) {

len = g[v][to];

if (d[v] + len < d[to]) {

d[to] = d[v] + len;

}

}

}

}

}

int main()

{

ifstream myFile("D:\\Learning\\ОТА\\4 семестр\\Лабораторные работы\\ЛР4\\ShortestWayInGraph\\ShortestWayInGraph\\Debug\\file.txt");

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

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

myFile >> g[i][j];

if (g[i][j] == -1) g[i][j] = INF;

}

}

clock\_t startTime = clock();

solve\_d();

double timeTaken = double(clock() - startTime) / (double)CLOCKS\_PER\_SEC;

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

cout << d[i] << " ";

}

cout << endl;

cout << "Time taken: " << timeTaken << " seconds." << endl;

}

* 1. РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММ
     1. Результат работы программы поиска кратчайших путей методом динамического программирования:

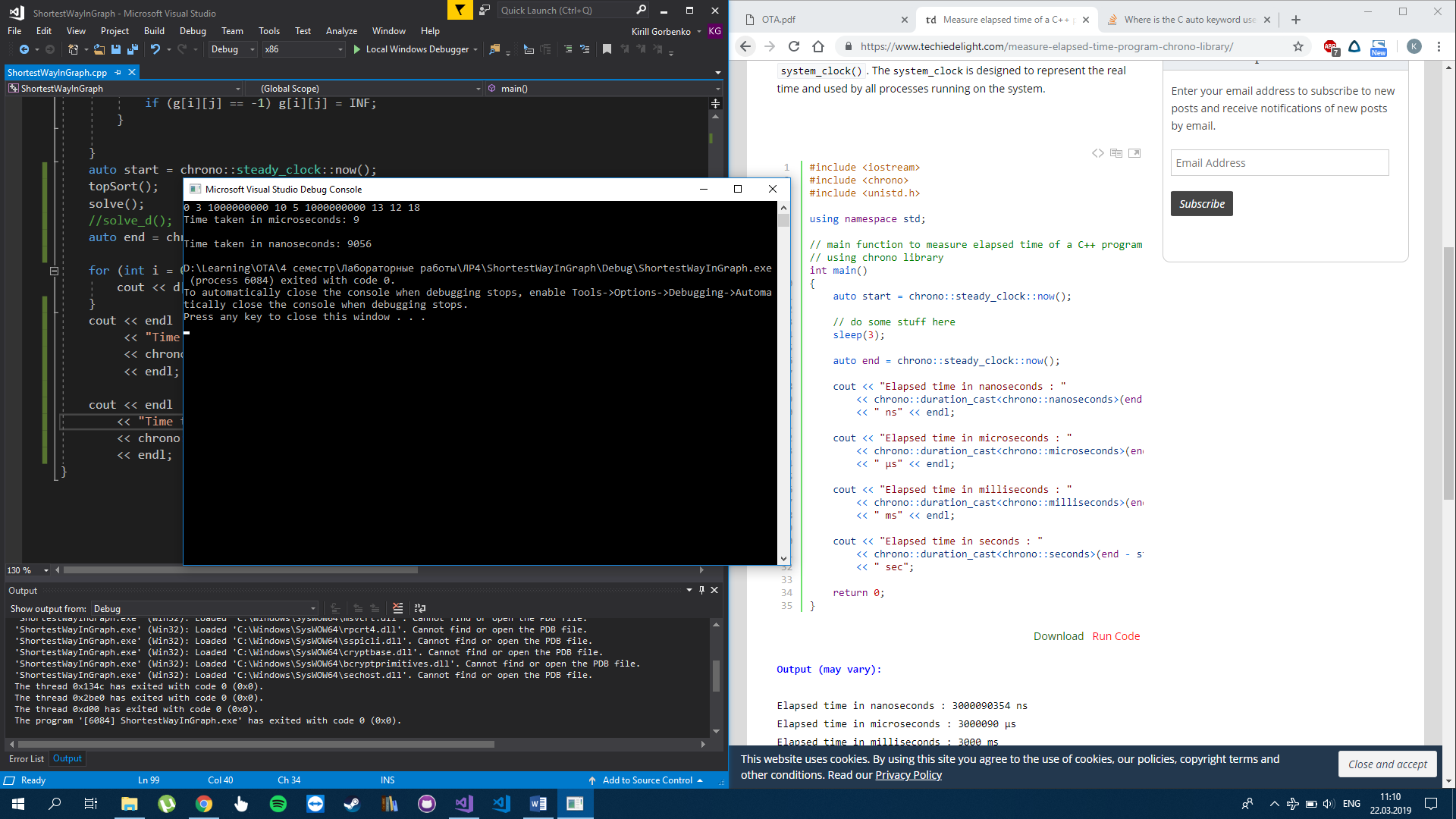


Рис. 5 – Результат работы программы поиска кратчайших путей методом динамического программирования

* + 1. Результат работы программы поиска кратчайших путей с использованием алгоритма Дейкстры изображен на рисунке № 6:

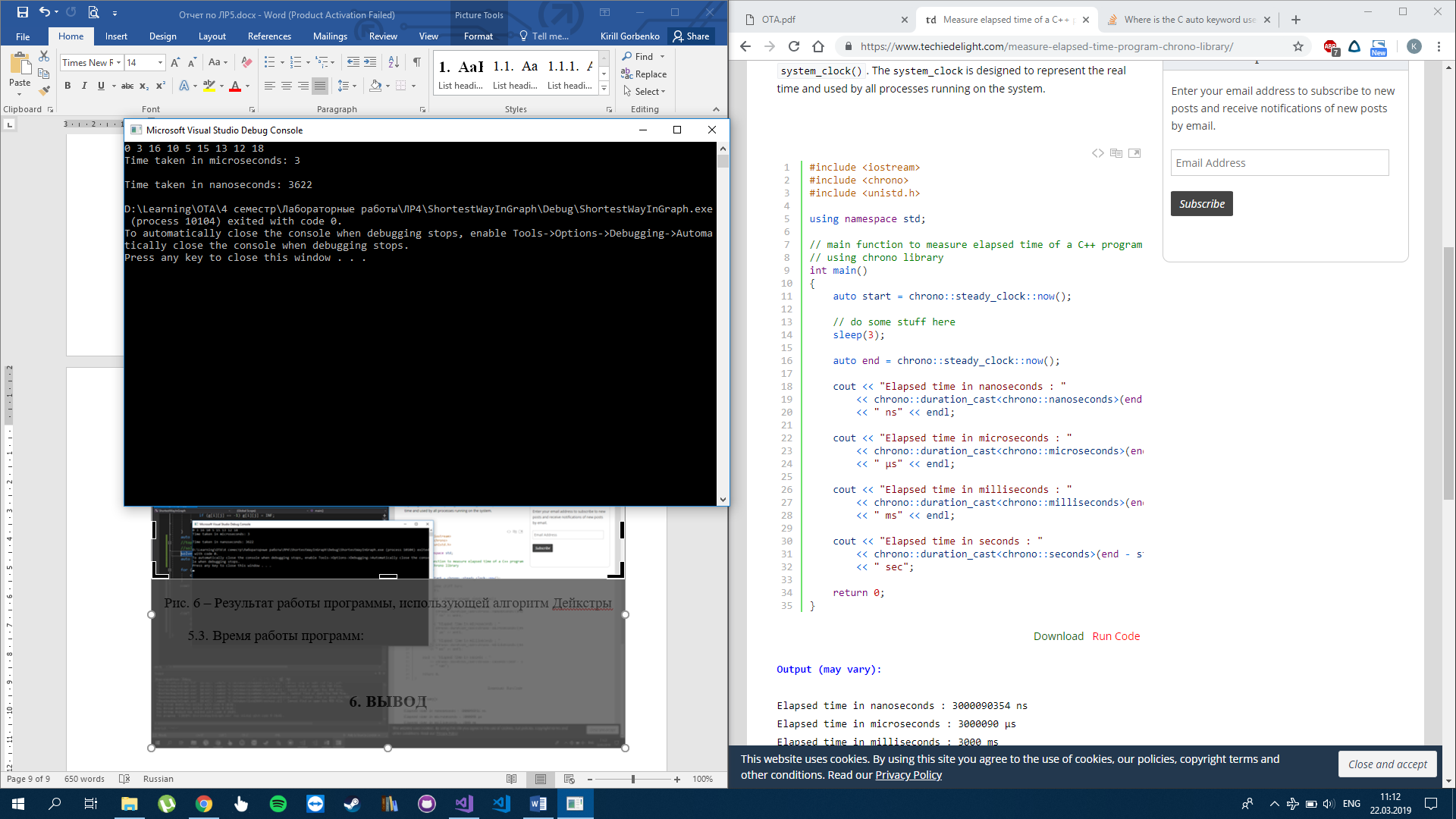


Рис. 6 – Результат работы программы, использующей алгоритм Дейкстры

* + 1. Время работы программ:

Программа поиска кратчайших путей методом динамического программирования завершилась за 9056 наносекунд; программа, использующая метод Дейкстры – за 3622 наносекунд. Алгоритм Дейкстры сработал в 9056 / 3622 = 2.5 раза быстрее.

* 1. ВЫВОД

В ходе лабораторной работы была решена задача о поиске кратчайших путей на графе. Для этого использовались два алгоритма: динамического программирования и Дейкстры. В результате получили идентичные ответы, однако, время выполнения программы, использующей алгоритм Дейкстры, оказалось в 2.5 раза быстрее. Кроме того, метод динамического программирования накладывает на входной граф ограничения в виде правильно подобранной нумерации, чего не требует алгоритм Дейкстры. В итоге, боллее предпочтительным является алгоритм Дейкстры.