**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе № 3**

**по дисциплине «Построение и Анализ Алгоритмов»**

**Тема: ВЫЧИСЛЕНИЕ РАССТОЯНИЙ МЕЖДУ ВСЕМИ ПАРАМИ ВЕРШИН – МЕТОД ФЛОЙДА**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Иванов В.С. |
| Преподаватель |  | Балтрашевич В.Э. |

Санкт-Петербург

2018

# **Цель работы**

Решение индивидуальной задачи посредством реализации алгоритма Флойда для вычисления расстояний между всеми парами вершин в графе.

# **Ход работы**

1. Интерфейс программы:

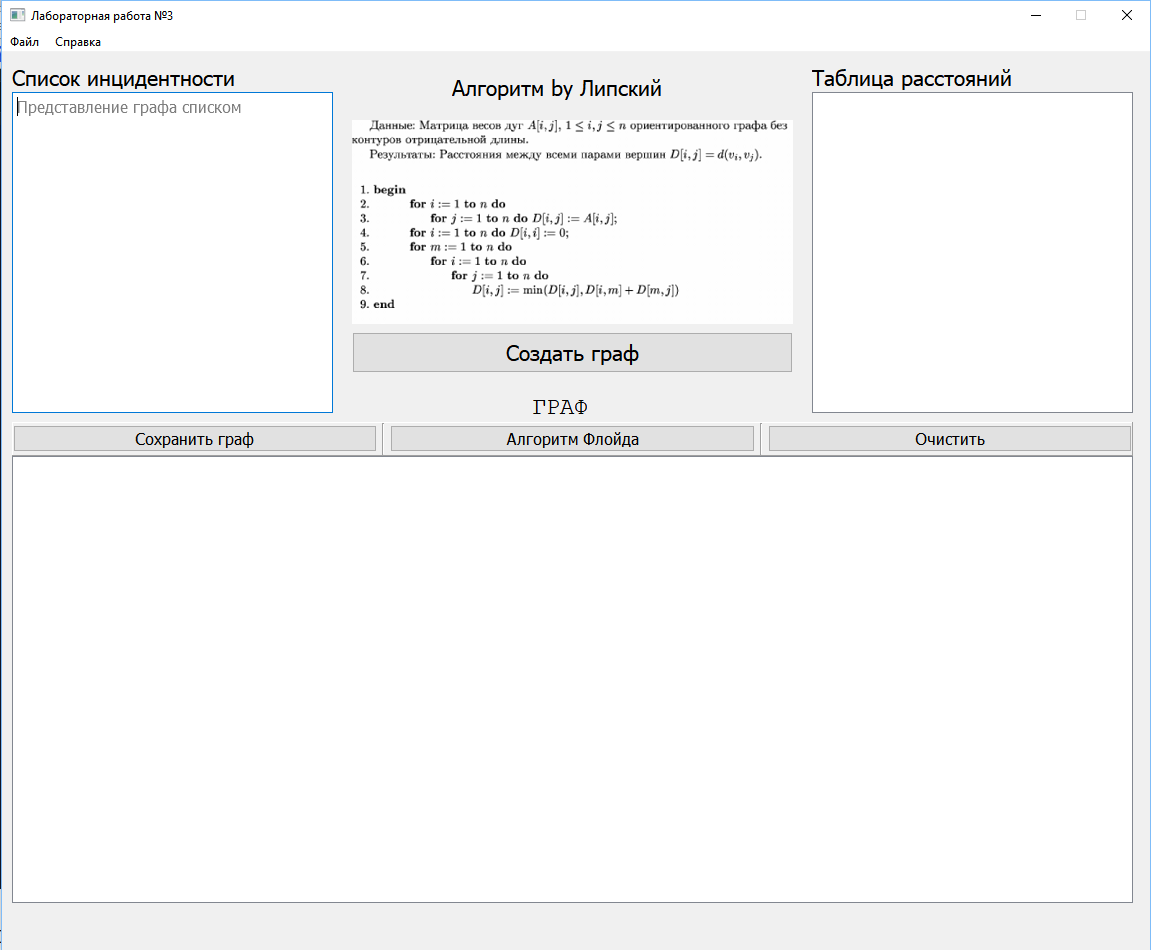


Рисунок 1 – Интерфейс программы

1. Очевидно, что задачу определения расстояния между всеми парами вершин можно решить, используя n раз (поочередно для каждой вершины) один из методов нахождения расстояний от фиксированной вершины. Таким образом, получается алгоритм со сложностью О(n^4) (при использовании метода Форда-Беллмана) или О(n^3) для бесконтурных графов. Однако многократное использование метода Форда-Беллмана не является наилучшим методом. Для данной задачи существует более эффективный алгоритм, созданный Флойдом:

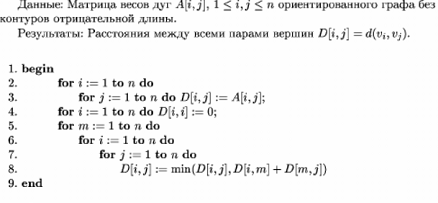


Рисунок 2 – Алгоритм из Липского

1. Сначала в матрицу записываются уже известные веса, при этом для элементов главной диагонали задаёт нули, а при отсутствии пути ставится знак бесконечности. Далее алгоритм пробегается по всем вершинам и ищет более короткий путь через вершину m (вершину, находящуюся в рассмотрении в данный момент), и если для какой-нибудь пары рёбер текущее расстояние между ними больше, чем путь, проходящий через вершину m, то значение этого пути записывается в матрицу смежности.
2. Реализация данного алгоритма:

for(int i = 0; i < size && processing; i++)

{

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

matrix[i][j] = INF;

matrix[i][i] = 0;

}

//первоначальное заполнение матрицы (по значениям рёбер между этими вершинами)

foreach (Node\* node, \*getMscene()->nodesList()) {

int a = getMscene()->nodesList()->indexOf(node);

foreach (Node\* ch, \*node->children()) {

int b = getMscene()->nodesList()->indexOf(ch);

Edge\* e= getMscene()->findEdge(node,ch);

matrix[a][b] = e->getValue();

}

}

//сам алгоритм, сделанный по Липскому

for(int k = 0; k < size && processing; ++k){

for(int i = 0; i <size && processing; ++i){

for(int j = 0; j < size && processing; ++j){

matrix[i][j] = qMin(matrix[i][j],matrix[i][k] + matrix[k][j]);

}

}

}

1. Таким образом реализован алгоритм для вычисления расстояний между всеми парами вершин в графе.

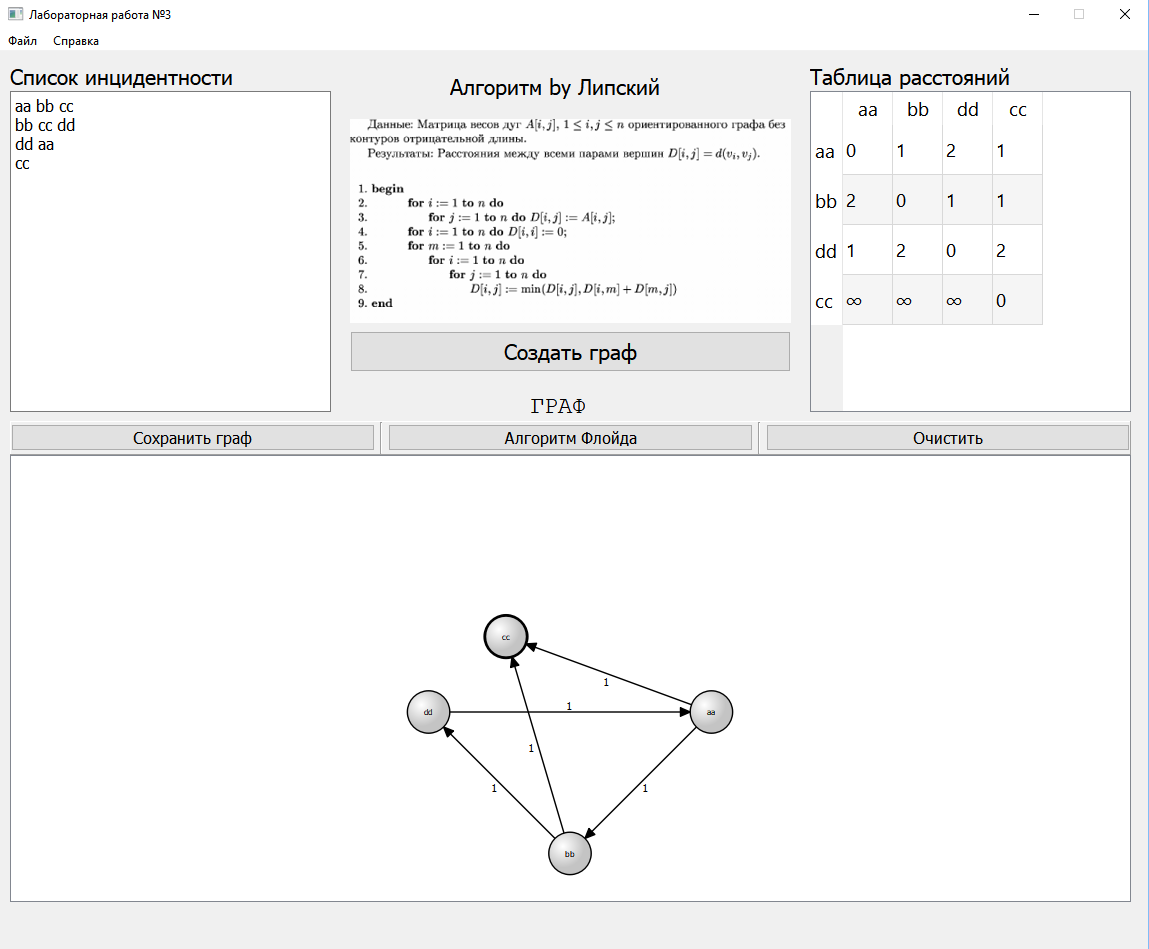


Рисунок 3 – Визуализация работы программы

# **Вывод**

В ходе выполнения данной лабораторной работы разработан и изучен алгоритм Флойда для вычисления расстояний между всеми парами вершин в графе.