

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по учебной практике
Тема: Минимальное остовное дерево. Алгоритм Краскала

Студент гр. 1304	_____	Байков Е.С.
Студент гр. 1303	_____	Депрейс А.С.
Студент гр. 1304	_____	Маркуш А.Е.
Руководитель	_____	Токарев А.П.

Санкт-Петербург
2023

ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ

Студент Байков Е.С. группы 1304

Студент Депрейс А.С. группы 1303

Студент Маркуш А.Е. группы 1304

Тема практики: Минимальное остовное дерево. Алгоритм Краскала

Задание на практику:

Командная итеративная разработка визуализатора алгоритма на Java с графическим интерфейсом.

Алгоритм: Минимальное остовное дерево. Алгоритм Краскала

Сроки прохождения практики: 30.06.2020 – 13.07.2020

Дата сдачи отчета: 00.07.2020

Дата защиты отчета: 00.07.2020

Студент гр. 1304	_____	Байков Е.С.
Студент гр. 1303	_____	Депрейс А.С.
Студент гр. 1304	_____	Маркуш А.Е.
Руководитель	_____	Токарев А.П.

АННОТАЦИЯ

Задачей учебной практики является получение опыта командной работы над проектом. Практика заключается в разработке приложения — визуализатора указанного в задании алгоритма на языке Java. В данном случае предстоит визуализация алгоритма Краскала для поиска минимального остовного дерева. Разработка ведётся итеративно: студентами и их руководителем согласовывается спецификация, чётко определяющая предполагаемое поведение программы, после чего создаются несколько версий проекта, каждая из которых расширяет функционал предыдущей. В отчёте приведена информация о ходе выполнения практической работы.

SUMMARY

The objective of the training practice is to gain experience in teamwork on a project. The practice consists in developing an application - visualizing the congestion of the algorithm specified in the task in the Java language. In this case, it is necessary to visualize the Kraskal algorithm for finding the minimum spanning tree. The development is carried out iteratively: students and their supervisor agree on a specification that clearly defines the intended behavior of the program, after which several versions of the project are created, each of which extends the functionality of the previous one. The report provides information on the progress of practical work.

СОДЕРЖАНИЕ

	Введение	5
1.	Требования к программе	6
1.1.	Исходные требования к программе	6
1.1.1.	Требования к визуализации работы алгоритма	6
1.1.2	Требования к визуализации пользовательского интерфейса	6
1.1.3	Требования к входным данным	8
1.2.	Уточнение требований после сдачи прототипа	0
1.3.	Уточнение требований после сдачи 1-ой версии	0
1.4.	Уточнение требований после сдачи 2-ой версии	0
2.	План разработки и распределение ролей в бригаде	9
2.1.	План разработки	9
3.	Особенности реализации	0
3.1.	Структуры данных	0
3.2.	Основные методы	0
3.3		0
4.	Тестирование	0
4.1	Тестирование графического интерфейса	0
4.2	Тестирование кода алгоритма	0
	Заключение	0
	Список использованных источников	0
	Приложение А. Исходный код – только в электронном виде	0

ВВЕДЕНИЕ

Цель практики – визуализация предоставленного алгоритма на языке Java.

Для выполнения цели практики следует решить следующие задачи:

1. Изучить алгоритм, который предстоит реализовать.
2. Составить и согласовать спецификацию разработки приложения.
3. Составить план разработки и распределить роли при реализации приложения.
4. Написать и отладить модули проекта.
5. Скомпилировать модули в один проект и отладить полное приложение.
6. Предоставить проект руководителю.

Алгоритм Краскала – эффективный алгоритм построения минимального остовного дерева взвешенного связного неориентированного графа. Алгоритм работает за линейное время на отсортированном множестве рёбер, а если учесть, что сортировка имеет сложность $O(E \times \log(E))$, то общее время работы алгоритма будет $O(E \times \log(E))$.

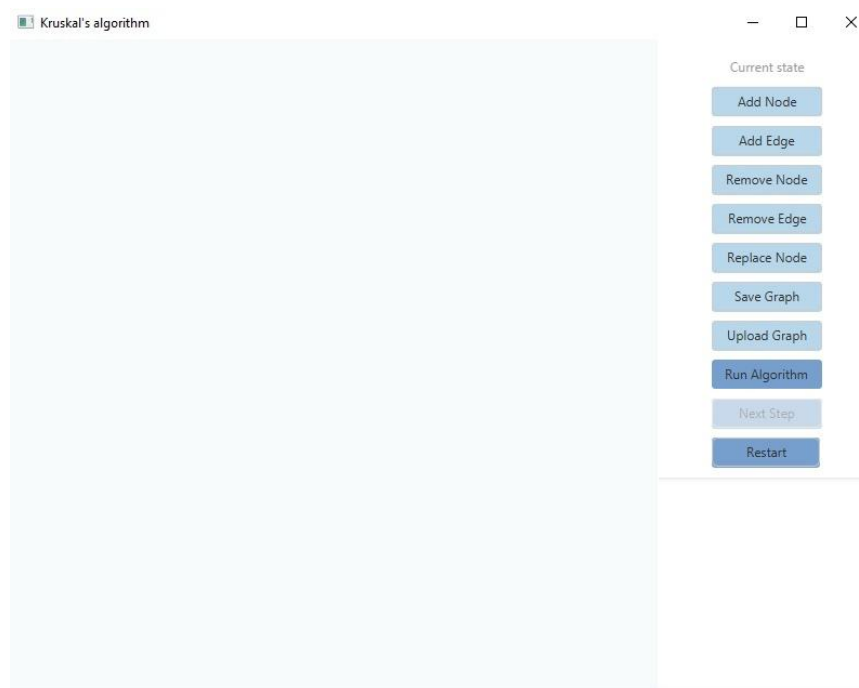


Рисунок 2 – Пользовательский интерфейс приложения в начале его работы

В начале запускается окно, в котором большую часть занимает зона для отрисовки, где будет находиться граф, справа находится панель управления с кнопками, в правом нижнем углу находится окно для предоставления информационных сообщений при исключениях, а так же при непосредственной работе алгоритма.

При создании графа, кнопка “Next Step” заблокирована, т.к. алгоритм ещё не начал своей работы. Кнопка “Save Graph” позволяет сохранить граф в файл текстовый файл. Кнопки “Add Node”, “Add Edge” переводят программу в режим добавления вершин либо рёбер соответственно, т.е. при нажатии на рабочее поле в режиме добавления вершин будет появляться вершина в указанной точке, а при нажатии на две вершины в режиме добавления рёбер будет появляться ребро между выбранными вершинами. Кнопки “Remove Node” и “Remove Edge” переводят программу в режим удаления вершин или рёбер. При нахождении в режиме удаления нажатие на объект соответствующий режиму приведёт к удалению этого объекта. При удалении вершины, все инцидентные ей рёбра тоже удаляются. Формат представления графа в файле будет представлен в п.1.1.3. После того, как пользователь нажмёт кнопку “Run Algorithm”, кнопки связанные с созданием и сохранением графа

будут заблокированы для пользователя, а кнопка “Next Step” станет доступной. Это можно увидеть на рис. 2.

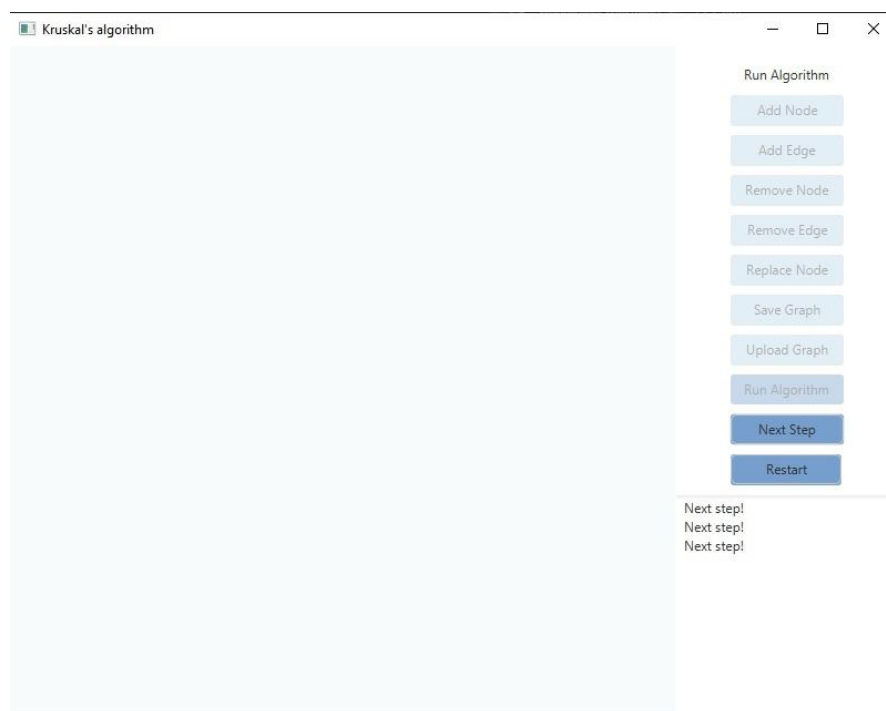


Рисунок 3 – Пользовательский интерфейс, после начала работы алгоритма
Кнопка “Restart” доступна всегда, она возвращает программу в начальное состояние как на рис. 2.

Как можно заметить на рис. 3 при работе алгоритма в окне сообщений выводится информация о пройденном шаге алгоритма. Вместо сообщения “Next step!” будет выводиться сообщение, поясняющее данный шаг алгоритма.

1.1.3. Требования к входным данным

Представления графа может осуществляться двумя способами:

1. Загрузка графа из .txt файла, по нажатию кнопки “Upload File”, в котором задано число вершин графа и построчно записана матрица смежности.
2. Создания графа с помощью инструментов, доступных в пользовательском интерфейсе.

1.2. Уточнение требований

2. ПЛАН РАЗРАБОТКИ И РАСПРЕДЕЛЕНИЕ РОЛЕЙ В БРИГАДЕ

2.1. План разработки и распределение ролей в бригаде

1. Составление спецификации. Исполнители: Маркуш А.Е., Депрейс А.С., Байков Е.С. Срок сдачи: 06.07
2. Составление плана разработки. Исполнители: Маркуш А.Е., Депрейс, А.С. Байков Е.С. Срок сдачи: 06.07
3. Реализация графического интерфейса:
 - а) Реализация области отображения графа. Исполнитель: Байков Е.С. Срок сдачи: 08.07
 - б) Реализация графического интерфейса для редактирование графа и работы алгоритма. Исполнитель: Байков Е.С. Срок сдачи: 08.07
4. Реализация работы с файлом:
 - а) Реализация сохранения графа в файл. Исполнитель: Маркуш А.Е. Срок сдачи: 08.07
 - б) Реализация загрузки графа из файла. Исполнитель: Маркуш А.Е. Срок сдачи: 08.07
5. Реализация бизнес логики программы.
 - а) Реализация алгоритма Краскала. Исполнитель: Депрейс А.С. Срок сдачи: 08.07
 - б) Реализация класса графа и его создания. Исполнитель: Депрейс А.С. Срок сдачи: 08.07
 - с) Реализация классов для вершины и дуги. Исполнитель: Депрейс А.С. Срок сдачи: 08.07
 - д) Создание объектов для передачи сообщений о работе алгоритма. Исполнитель: Депрейс А.С. Срок сдачи: 08.07
6. Реализация связующих структур между графической и логической частью программы. Исполнители: Маркуш А.Е., Депрейс А.С. Срок сдачи: 10.07
7. Реализация визуализации работы алгоритма:

- a) Добавление возможности запуска алгоритма и просмотра каждого его шага. Исполнитель: Байков Е.С. Срок сдачи: 10.07
- b) Реализация API для связи с логикой программы. Исполнитель: Байков Е.С. Срок сдачи: 10.07

8. Внесение корректировок, исправление багов и тестирование программы:

- a) Тестирование и исправление графической части программы. Исполнитель: Байков Е.С. Срок сдачи: 12.07
- b) Тестирование и исправление сохранения/загрузки графа. Исполнитель: Маркуш А.Е. Срок сдачи: 12.07
- c) Тестирование и исправление бизнес логики программы. Исполнитель: Депрейс А.С. Срок сдачи: 12.07

3. ОСОБЕННОСТИ РЕАЛИЗАЦИИ

3.1. Структуры данных

3.2. Основные методы

4. ТЕСТИРОВАНИЕ

4.1. Первый подраздел третьего раздела

4.2. Второй подраздел третьего раздела

ЗАКЛЮЧЕНИЕ

Кратко подвести итоги, проанализировать соответствие поставленной цели и полученного результата.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Java документация от Oracle // Oracle. URL:
2. Metanit. URL: <https://metanit.com/java/javafx>
3. Stackoverflow. URL: <https://stackoverflow.com/>
4. Wikipedia. URL: https://ru.wikipedia.org/wiki/Шаблон_проектирования
5. Habr. URL: <https://habr.com/ru/articles/474982/>

ПРИЛОЖЕНИЕ А
НАЗВАНИЕ ПРИЛОЖЕНИЯ