­­­­­­

Содержание

[ВВЕДЕНИЕ 2](#_Toc9767693)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 4](#_Toc9767694)

[1.1 Анализ аналогов программного средства 4](#_Toc9767695)

[1.1.1 Онлайн сервис работы с графами https://graphonline.ru 4](#_Toc9767696)

[1.1.2 Визуализатор графов https://csacademy.com/app/graph\_editor/ 4](#_Toc9767697)

[1.2 Силовые методы укладки графов 5](#_Toc9767700)

[1.3 Алгоритм Дейкстры 5](#_Toc9767703)

[2 Постановка задачи 6](#_Toc9767708)

[3 РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА 7](#_Toc9767709)

[3.1 Разработка основной логики программы 7](#_Toc9767710)

[3.2 Алгоритм, использующий силовые методы 7](#_Toc9767714)

[3.3 Требования к программе 1](#_Toc9767715)0

[4 ДЕМОНСТРАЦИЯ РАБОТЫ 1](#_Toc9767716)1

[ЗАКЛЮЧЕНИЕ 1](#_Toc9767717)5

[ПРИЛОЖЕНИЕ....................................................................................................1](#_Toc9767717)6

# 

# ВВЕДЕНИЕ

В настоящее время цифровые технологии все больше проникают в сферу образования. Основной задачей таких технологий становится улучшить качество образования за счет того, что они способны обеспечить наглядность, интерактивность обучающего процесса. Мультимедийные средства, такие как анимация, видео и изображение, позволяют упростить восприятие информации и отобразить сложные процессы и явления в деталях. Такие технологии являются незаменимым помощником преподавателей, желающих предоставить материал в наиболее понятной форме, а также оставить часть его в качестве материала для самостоятельного обучения.

Особую роль цифровые технологии играют в самообразовании. Современные онлайн-курсы полностью строятся на медиа-контенте и его интерактивности, подача материала может осуществляться в виде программы, где пользователь сам выбирает различные параметры и условия процесса. Такой подход обладает рядом плюсов: изучение материала при помощи наглядных примеров; возможность изменить процесс, для того, чтобы пользователь понимал, правильно ли он усвоил материал; независимость процесса изучения от преподавателя; активное участие пользователя в процессе обучения.

Несмотря на преимущество использования такого метода, массово в белорусском школьном образовании он не используется. Потенциальное использование таких технологий смогло бы не только повысить эффективность усвоения обязательной программы, но дать возможность в дополнительном углубленном изучение материала как факультативно, так и в рамках самообразования.

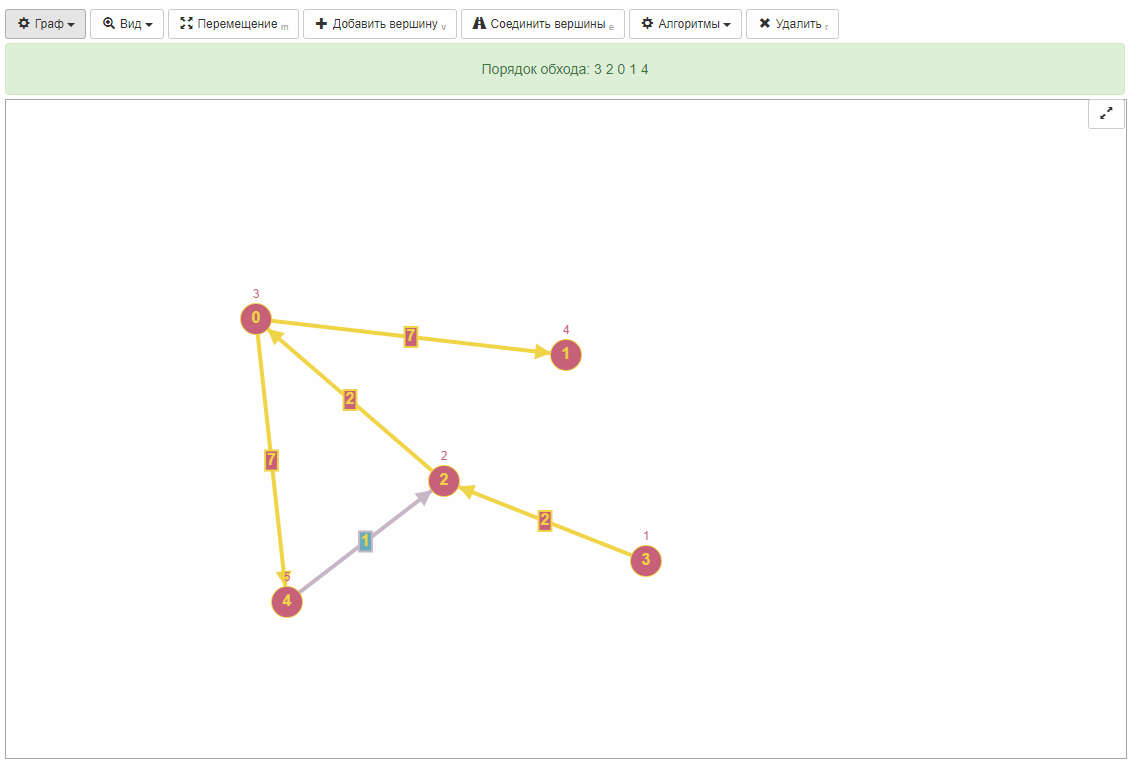
Наряду с этим, во многих школах остро стоит проблема в углубленном изучении информатики. Современная программа направлена на основы работы с информационными технологиями, тогда как многим школьникам, планирующим поступление на специальности, связанные с программированием, необходимо базовые знания алгоритмов и структур данных. Поэтому во многом школьное образование не дает основ в данном направлении, что вызывает потребность изучения материала либо факультативно, либо самостоятельно. Особенности использования интерактивных методов изучения позволят разрешить данную проблему.

В качестве одного из направлений изучения выбраны алгоритмы на графах – одна из основных тем многих курсов по алгоритмам и структурам данных. Программа, позволяющая быстро сгенерировать и построить граф, тем самым повысив эффективность процесса обучения, а также визуализировать результаты работы алгоритмов, стала бы главной опорой в изучении данной темы как в пределах факультативного курса в школе, так и самостоятельного рассмотрения.

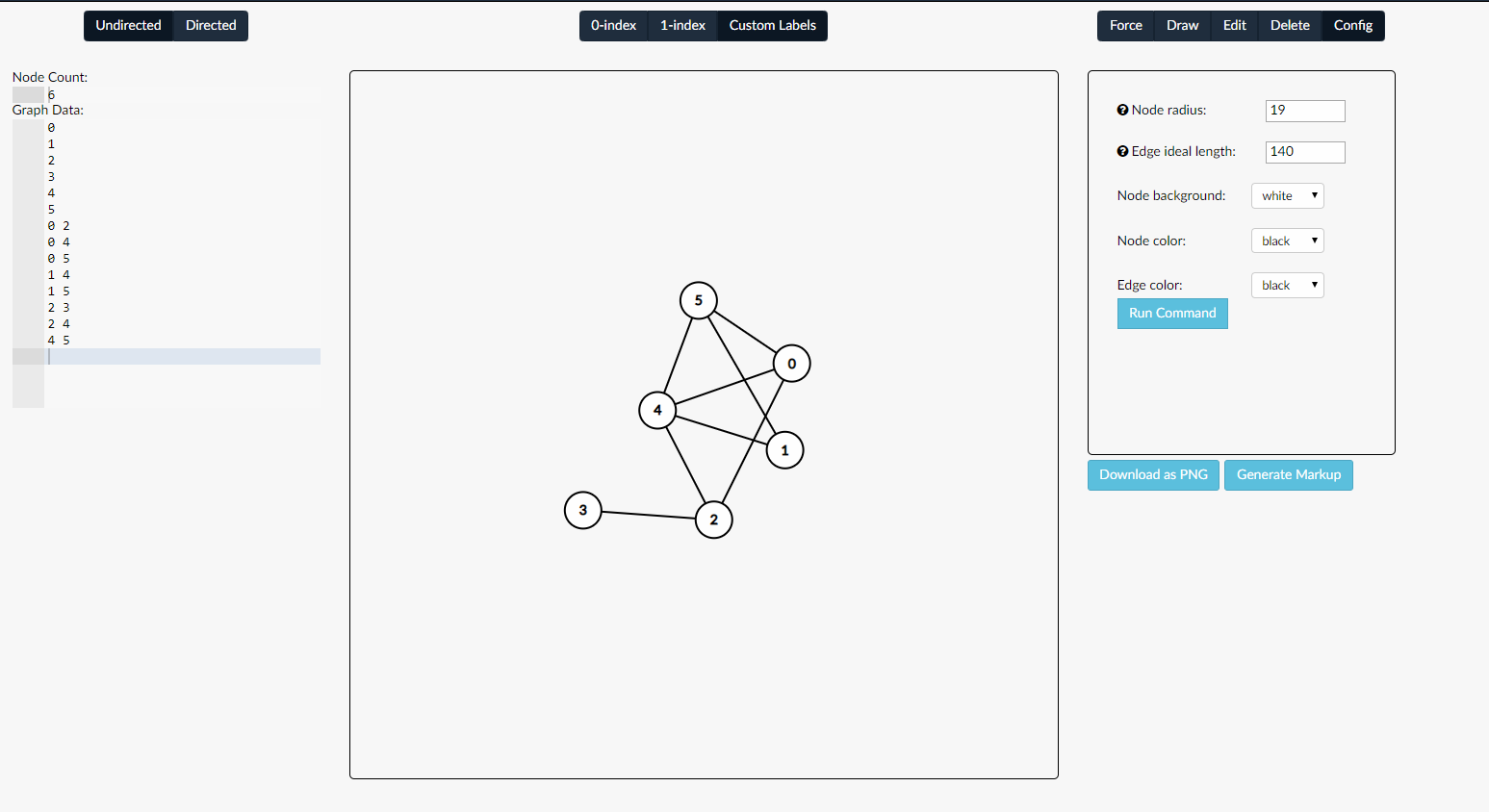
# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

# Анализ аналогов программного средства

### Онлайн сервис работы с графами <https://graphonline.ru>

 Данный сервис является удобным инструментом для визуализации алгоритмов на графах. Сайт обладает широким функционалом, позволяющим быстро и удобно построить любой вид графа вручную, загрузить или сохранить матрицу смежности графа. Также на сайте представлено множество алгоритмов, многие из которых при визуализации сопровождаются анимацией работы.

### Визуализатор графов <https://csacademy.com/app/graph_editor/>

Данное приложение предназначено для визуализации силового метода укладки графов на плоскости. Отличительной особенностью данного приложения является анимация этапов работы силового алгоритма и возможность настройки таких визуальных элементов как радиус вершин, цвет вершин и ребер. 

### **1.2 Силовые методы укладки графов**

Алгоритм, основанный на симуляции физической модели позволяет произвести укладку графа на плоскости. Из очевидных плюсов простота в реализации и универсальность алгоритма, не требующая дополнительных свойств графа. Минусом является недетерминированность алгоритма и сложность в подборе макропараметров, что влечет за собой вероятность получения неупорядоченного графа. При неправильном подборе макропараметров возможно зацикливание работы алгоритма.

### **1.3 Алгоритм Дейкстры**

Один из базовых алгоритмов на графах, представляющий собой поиск наикратчайшего расстояния между двумя вершинами графа. Результатом работы такого алгоритма является массив предков пути из каждой вершины и кратчайшее расстояние от стартовой вершины до каждой вершины графа. Визуализация алгоритма возможно только на взвешенных неориентированных графах.

### **Постановка задачи**

Цель: применение теоретических и практических знаний для создания программы-визуализатора графов и алгоритмов на них.

В данном проекте главными задачами станут:

* Создание удобного пользовательского интерфейса для взаимодействия со всеми функциями проекта
* Возможность быстро нарисовать граф вручную, чтобы пользователь мог визуализировать любую нужную конфигурацию графа
* Визуализация результата алгоритма на данном графе

### **РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА**

### **Разработка основной логики программы**

При запуске программы пользователь может выбрать метод создания нового графа: создание нового графа рисованием, или же задавать граф списком рёбер.

Затем на основании этих данных будет построен граф. В любой момент выполнения программы пользователь может перемещать граф для лучшего визуального представления.

Пользователь может в любой момент перестроить граф и начтать выполнения алгоритма Дейкстры.

Основой визуализации стал класс Draw\_form, для отображения графа был выбран объект PIctureBox который поддерживает рисование объектов на канве. При помощи SetNewPoint, pictureBox\_Paint, pictureBox\_MouseDown, pictureBox\_MouseUp и pictureBox\_MouseMove происходит взаимодействие с формой, отображающей граф. Реализовано перемещение по канве, создание вершин и ребер, просмотр информации о вершине, рисование и выбор вершин как параметров алгоритма.

Вся логика работы алгоритма Дейкстры содержится в классе Deextra\_setting\_form.

Все данные хранятся в классе MyData В качестве полей класс включает в себя логические переменные, описывающие свойства графа; массив номеров вершин, соединенных этим ребром; список смежности данного графа;

### **3.2 Алгоритм, использующий силовые методы**

Многие сложные задачи в программирования решаются при помощи эвристических алгоритмов. Эвристическими алгоритмы – класс алгоритмов, включающих в себя практический метод, не являющийся гарантированно точным или оптимальным, но достаточный для решения поставленной задачи.

Одним из эвристических методов является построении алгоритма решения задачи на основе физических моделей, описывающих определенный процесс.

Одним из методов укладки графов, основанном на физических моделях, является Force-Directed Graph Drawing – силовой алгоритм рисования графов. Главой идеей данной эвристики является физический факт – любая система стремится к состоянию минимальной энергии, в котором минимум дефектов и наблюдается симметрия.

В качестве физической системы выбрана следующая модель: пусть в каждой вершине графа помещены заряды величиной , а ребра представляют собой пружины жесткости и длинной . Данная система случайным образом размещается на плоскости, каждой вершине генерируется случайная скорость . Затем система изменяется так, чтобы прийти в состояние покоя. Для этого необходим процесс релаксации: на каждую вершину действует сила трения с постоянной . Состояние покоя можно считать ту конфигурацию системы, когда ее кинетическая энергии достигла нулевого или почти нулевого значения. Следовательно, задачей становится нахождение минимума функции . В алгоритме использовался метод градиентного спуска для нахождения минимума данной функции.

Для реализации градиентного спуска требуются производные – положения точки и – скорости данной точки: , , где – сила, действующая на точку в данный момент времени. Тогда, изменение состояния системы примет вид: и , где – постоянная градиентного спуска.

В данной физической модели на вершину действует суперпозиция трех сил:, электростатической силы, силы Гука и силы трения: , , ,

.

К плюсам такого алгоритма можно отнести относительную наглядность графа при большом количестве ребер и вершин, чем в наивных методах укладки графов.

Основным минусом такого метода являются случайность полученного результата, так как алгоритм эвристический, а, следовательно, возможно неоптимальное решение укладки графа: большое количество пересечений ребер и вершин, несимметричность, большая площадь графа. Также возникает сложность с подбором макропараметров , так как функция может не достигнуть своего глобального минимума, а наоборот уйти в локальный максимум или же достичь локального минимума, при котором кинетическая энергия будет меньше начальной, но не будет нулевой.

### 

### **3.3 Требования к программе**

К проекту предъявлены следующие требования

1. Интерфейс

Интерфейс программного средства должен содержать рабочую область, на которой будет расположен граф.

1. Функциональность

Пользователю предоставлены различные способы построения графа.

1. Наглядность

должен быть удобным и понятным для пользователя

1. Среда разработки

Средой разработки данного программного средства выбран Visual Studio

### **ДЕМОНСТРАЦИЯ РАБОТЫ**

При запуске нас встречает главное меню где можно выбрать способ задания графа, а так же запуск алгоритма Дейкстры.

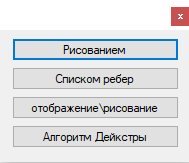


Рисунок 4.1 Главное меню программы

При выборе Рисования откроется меню рисования а так же поле для отображения

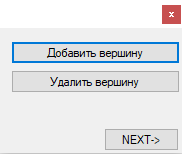


Рисунок 4.2.1 Меню рисования графа

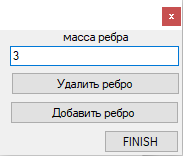


Рисунок 4.2.2 Меню рисования графа

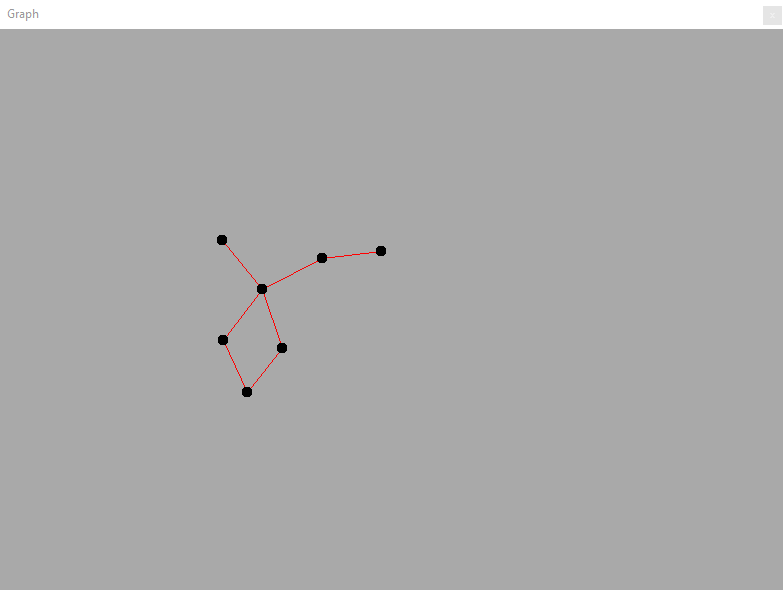


Рисунок 4.3 поле отображения графа

При выборе режима задания списком рёбер открывается меню списка рёбер.

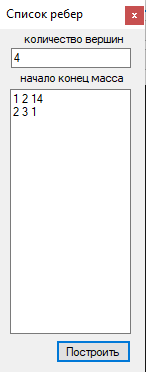


Рисунок 4.4 Меню списка рёбер

При запуске алгоритма Дейкстры, запускается меню.

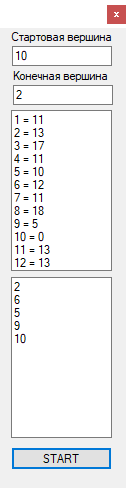


Рисунок 4.4 Меню алгоритма

При визуализации алгоритма закрашиваются вершины, ребра.

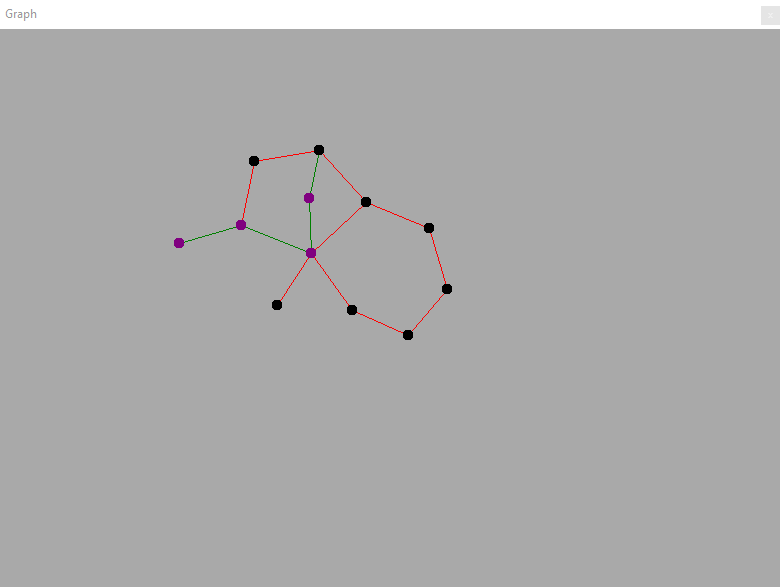


Рисунок 4.4 Визуализация алгоритма максимального потока

### **ЗАКЛЮЧЕНИЕ**

В результате работы над алгоритмом было разработана программа, которая позволяет удобно и быстро построить граф, а также визуализировать результат работы алгоритма на графах.

Плюсами данного программного средства можно считать удобный и понятный интерфейс, широкий функционал, позволяющий выбрать пользователю наиболее подходящие для работы инструменты. Стоит отметить архитектуру проекта, которая позволяет быстро добавлять к проекту необходимые функции.

В качестве языка разработки был выбран С# и фреймворк Visual Studio, так как библиотека алгоритмов, на которой основан данный проект была реализована именно на этом языке программирования.

Разработанное программное средство представляет собой как законченный проект с полностью реализованным функционалом, так и основу для развития более совершенного программного средства, которое сможет включать в себе не только алгоритмы на графах.

**ПРИЛОЖЕНИЕ**

Исходный код https://github.com/redboostik/kursach/tree/master/GraphViewer