

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ «АЛГОРИТМЫ НА ГРАФАХ»
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы

направления 02.03.03 Математическое обеспечение и администрирование
информационных систем

факультета компьютерных наук и информационных технологий

Исайкина Виктора Викторовича

Научный руководитель:

зав.кафедрой, к.ф.м.н

_____ М.В. Огнева

Зав. кафедрой:

к.ф.-м.н.

_____ М.В. Огнева

Саратов 2017

ВВЕДЕНИЕ

Граф — математическая модель для представления объектов или событий и связей между ними. Простой пример графа — схема метрополитена, где в качестве вершин выступают станции, а в качестве ребер — связи между станциями. Графы используют в самых разнообразных областях науки и практики. Структуры молекул, блок-схемы алгоритмов, электрические цепи, схемы дорог, предприятия со связями между ними, группы людей с указанием каких-либо отношений (например, психологической совместимости), структура управления с указанием объектов — это и многое другое можно представить с помощью графов [1].

Использование методов теории графов часто ускоряет решение практических задач, упрощает расчеты, повышает эффективность научной, инженерной и конструкторской деятельности. Возьмем схему метрополитена Москвы. Если посмотреть на нее, то нетрудно дать ответ на вопрос: «Сколько станций между Парком культуры и Бауманская». Это простая задача и не требует значительных усилий для решения. Однако если количество станций будет порядка 10^3 , то задача становится затруднительной для человека, но довольно просто решается алгоритмом поиска кратчайших путей на не взвешенном графе [1].

Задачи на графы встречаются еще в младшей школе, на уроках математики и информатики, хотя понятие "граф" еще не вводится. В старшей школе с помощью теории графов решаются уже достаточно серьезные задачи, в том числе, олимпиадные. В вузе теория графов изучается как в курсе информатики, дискретной математики, так и в виде отдельных дисциплин на направлениях и специальностях, связанных с компьютерными науками и математикой [1]. Студенты факультетов, связанных с компьютерными науками, "встречаются" с графовыми алгоритмами на олимпиадах, на практике, при написании курсовых и выпускных квалификационных работ, а профессиональные разработчики все чаще и чаще сталкиваются с необходимостью использовать алгоритмы на графах в

промышленных проектах. Основой теории графов являются методы обхода графа, на них основываются многие важные алгоритмы. Поэтому создание портала по изучению теории графов, который бы позволял размещать и редактировать теорию, добавлять задачи и тесты, отслеживать успеваемость и имел бы модуль визуализации графовых алгоритмов, является актуальным. Для облегчения поддержки портала необходимо сделать его кроссплатформенным, то есть не связанным со средой размещения.

Таким образом, целью данной работы является разработка и наполнение клиент-серверного приложения для обучения и контроля полученных знаний по теме «Теория графов. Алгоритмы обходов».

Для реализации поставленной цели необходимо:

1. Дать основные понятия теории графов
2. Представить описание и реализацию алгоритмов обходов графов, а также алгоритмов, основанных на обходах.
2. Рассмотреть инструменты для реализации клиент-серверного приложения для платформы .net core.
3. Разработать архитектуру приложения.
4. Реализовать модульное тестирование для проверки работоспособности компонентов.
5. Рассмотреть современные фреймворки для реализации клиентской части приложения.
6. Рассмотреть технологии доступа к данным.
7. Изучить методологию разработки tdd.
8. Разработать клиент-серверное приложение по теории графов.

В теоретической части работы представлены основные сведения о теории графов, и разобраны алгоритмы обходов, и алгоритмы, основанные на обходах.

В практической части было реализовано веб приложение по теории графов, позволяющее визуализировать алгоритмы на графах и контролировать усваиваемость теоретического материала.

Бакалаврская работа состоит из введения, пяти разделов, заключения, списка использованных источников и десяти приложений. Общий объем работы – сто двадцать две страницы, из них сорок четыре страницы – основное содержание, включая двадцать пять рисунков, цифровой носитель в качестве приложения, список использованных источников информации – двадцати двух наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Введение в теорию графов» посвящен основным понятиям теории графов. Раздел содержит необходимые определения и важные алгоритмы теории графов.

Содержание:

1. Основные понятия
2. Обходы графов
3. Поиск кратчайшего пути в невзвешенном графе
4. Проверка связности графа
5. Проверка графа на ацикличность
6. Проверка графа на двудольность

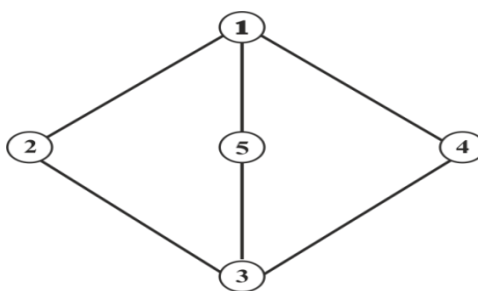


Рис.1.1. Пример графа

Графом $G = (V, E)$ называется пара двух множеств, где V — множество вершин, E — множество пар вершин. Если пары неупорядоченные, т.е. $(x, y) = (y, x)$ $x, y \in V$, то они называются ребрами, а граф называется неориентированным. В противном случае, если $(x, y) \neq (y, x)$ $x, y \in V$, то такие пары называются дугами, а граф — ориентированным. [2] Вершины u и v называются смежными, если они соединены ребром, т.е. существует $(u, v) \in E$. При этом говорят, что вершина u (или v) и ребро (u, v) инцидентны.

Путь — последовательность ребер(в неориентированном графе) и/или дуг (в ориентированном графе), такая, что конец одной дуги (ребра) является началом другой дуги (ребра) и никакое ребро не встречается два раза [2].

Граф называется связным, если для любых вершин v и u , существует путь из u в v .

Граф называется деревом, если он связный и не содержит циклов.

Цикл — замкнутый путь, в котором каждая вершина, за исключением начальной и конечной вершины, встречается один раз.

Способы представления

Для представления графов в компьютере чаще всего используют матрицы смежности или списки смежности.

Матрица смежности графа G с конечным числом вершин n — это квадратная матрица размера $n \times n$, в которой элемент $a_{i,j}$ на пересечении i строки и j столбца задается следующим образом:

$$a_{i,j} = \begin{cases} 1, & \text{существует ребро } (i,j) \\ 0, & \text{не существует ребро } (i,j) \end{cases}$$

Построим матрицу смежности для рисунка 1.1. В случае неплотных графов ($|E| < n^2$) матрица будет сильно разрежена, то есть большая часть памяти напрасно будет тратиться на хранение нулей [2]. Однако данный способ хранения позволяет быстро отвечать на вопрос о смежности вершин - для этого необходимо просто посмотреть на пересечение i строки и j столбца.

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Рис. 1.2. Матрица смежности графа на рис. 1.1

Список смежности графа с конечным числом вершин - это такой способ представления графа, когда для каждой вершины задается список смежных с ней вершин.

1: 2, 4, 5
2: 1, 3
3: 2, 4
4: 1, 3
5: 1, 3

Рис. 1.3. Список смежности графа на рисунке 1.1

Второй раздел «Разработка клиент-серверного приложения»

посвящен реализации веб-приложения по теории графов.

Содержание:

1. Платформа .NET Core
2. Сравнение с другими платформами .NET
3. ASP.NET Core

Основной упор сделан на выбор технологии, разработку архитектуры и тестирование функционала приложения. Был выбран фреймворк ASP.NET CORE MVC, так как является кроссплатформенным решением для разработки веб приложения на языке С#.

Третий раздел «Вспомогательные технологии»

Основной упор сделан на выбор архитектуры приложения, выбор вспомогательных технологий для разработки веб-приложения и выбор методологии разработки.

Содержание:

1. Трехслойная архитектура
2. Dapper
3. SQL server

Была выбрана трехслойная архитектура, фреймворк Dapper для облегчений слоя представления данных и методология разработки через тестирование.

Четвертый раздел «Реализация клиент-серверного приложения по теории графов»

Основной упор на разработку веб-приложения по теории графов.

Содержание:

1. Модуль с теоретическим материалом
2. Модуль визуализации
3. Вход и регистрация на портале
4. Модуль тестирования

Были разработаны следующие модули; модуль контроля знаний, позволяющий создавать и проходить тесты, модуль визуализации алгоритмов, позволяющий отрисовывать граф и визуализировать алгоритмы на графах, модуль с теоретическим материалом.

Пятый раздел «Тестирование клиент-серверного приложения»

Клиент-серверное приложение по теории графов было протестировано юнит тестами. Были созданы модульные тесты для слоя бизнес логики и доступа к данным, а также на сборку с валидацией. При реализации модульных тестов использовался фреймворк Моq, который позволяет эмулировать поведение объектов. Тесты находятся в приложении Л «Модульные тесты». Также были написаны интеграционные тесты для проверки корректности работы всей системы. Интеграционные тесты находятся в приложении М «Интеграционные тесты». При разработке была применена техника разработки через тестирование (Test-Driven Development).

Приложение успешно прошло все написанные тесты, поэтому можно утверждать, что приложение работает корректно.

ЗАКЛЮЧЕНИЕ

В данной работе было разработано клиент-серверное приложение для обучения и контроля полученных знаний по теме «Теория графов. Алгоритмы обходов». Приложение содержит теорию по данной теме, модуль визуализации и возможности для проведения тестирования.

Для реализации поставленной задачи были даны основные понятия теории графов, представлено описание и реализация алгоритмов обходов графов, а также алгоритмов, основанных на обходах, рассмотрены инструменты для реализации клиент-серверного приложения для платформы .net core, реализовано модульное тестирование для проверки работоспособности компонентов.

По тематике бакалаврской работы был представлен доклад: «Алгоритмы на графах: от школьных олимпиад до промышленных IT проектов» на III Международной научно-практической конференции «Электронное обучение в непрерывном образовании 2016», Ульяновск, УлГТУ, апрель 2016 г. Доклад опубликован в материалах конференции.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Огнева М.В, Исайкин В.В Алгоритмы на графах: от школьных олимпиад до промышленных IT проектов. Материалы сборника Электронное обучение в непрерывном образовании – Ульяновск: УлГТУ, 2016. – 1320 с.
- 2 Асанов М. О., Баранский В. А., Расин В. В. Дискретная математика: графы, матроиды, алгоритмы. — Ижевск: НИЦ «Регулярная и хаотическая динамика», 2001. — 288 с
- 3 Томас Х. Кормен, Чарльз И. Лейзерсон и др. Алгоритмы. Построение и анализ. Вильямс 2012 — 1299 с
- 4 Руководство по .NET Core. [Электронный ресурс]. — URL: <https://docs.microsoft.com/ru-ru/dotnet/core/> (Дата обращения 25.05.2017).
- 5 Введение в ASP.NET Core. [Электронный ресурс]. — URL <https://metanit.com/sharp/aspnet5/1.1.php> (Дата обращения 25.05.2017).
- 6 Много уровневая архитектура [Электронный ресурс]. — URL <https://metanit.com/sharp/mvc5/23.5.php> (Дата обращения 25.05.2017).
- 7 Тестирование веб приложений. [Электронный ресурс]. — URL <https://metanit.com/sharp/mvc5/18.1.php> (Дата обращения 25.05.2017)
- Фреймворк Моq. [Электронный ресурс]. — URL <https://metanit.com/sharp/mvc5/18.5.php> (Дата обращения 25.05.2017).
- 8 Кудрина Е.В., Огнева М.В. Теория графов: от школы до вуза. Материалы V Всероссийской научно-практической конференции «Информационные технологии в образовании». - Саратов: Издательский центр "Наука", 2013, С.23-26.
- 9 Графоанализатор [Электронный ресурс] URL: <http://grafoanalizator.unicksoft.ru> (дата обращения 30.05.2017)
- 10 GraphOnline [Электронный ресурс] URL: <http://graphonline.ru/> (дата обращения 30.05.2017)

- 11 Algo-visualize [Электронный ресурс] [URL:http://algo-visualizer.jasonpark.me/#path=graph_search/bfs/shortest_path](http://algo-visualizer.jasonpark.me/#path=graph_search/bfs/shortest_path) (дата обращения 30.05.2017)
- 12 Френк Харри. Теория графов. Либриком 2009. — 302 с.
- 13 Ойстин Оре. Теория графов. Либриком 2009. — 354 с.
- 14 Нина Костюкова. Графы и их применение. Комбинаторные алгоритмы для программистов. Интернет-университет информационных технологий, Бином. Лаборатория знаний 2016. — 312 с.
- 15 С.М. Окулов. Программирование в алгоритмах. Бином. Лаборатория знаний 2013. — 384 с
- 16 Седжвик Р. Фундаментальные алгоритмы на С++. Алгоритмы на графах: Пер. с англ. — СПб: ООО «ДиаСофтЮП», 2002. — 496 с.
- 17 Ахо А. и др. Структуры данных и алгоритмы: Пер с англ.: М.: Издательский дом «Вильямс», 2003 с. — 384 с.
- 18 Липский В. Комбинаторика для программистов: Перевод с польского — М.: «Мир», 1988. — 200 с.
- 19 Клейнберг Дж., Тардос Е. Алгоритмы: разработка и применение. Классика Computers Science / Пер. с англ. Е. Матвеева. — СПб.: Питер, 2016. — 800 с.
- 20 Роберт С. Мартин, Мика Мартин. Принципы, паттерны и методики гибкой разработки на языке С#. Символ-Плюс, 2011. — 768 с.
- 21 Джеффри Рихтер. CLR via С#. Программирование на платформе Microsoft.NET Framework 4.5 на языке С#. Питер, 2016. — 896 с.