

что указывает на переобучение модели. Значение AUC улучшается на протяжении обучения, достигая уровня 0,92–0,95. Это свидетельствует о том, что модель способна эффективно отделять классы, даже если точность остаётся низкой. Precision остаётся на уровне 0,6, что указывает на большое количество ложноположительных предсказаний. Recall достигает значения 1,0, что означает, что модель успешно идентифицирует все примеры с мутациями, жертвуя точностью. Loss на обучающей выборке снижается, что указывает на успешное обучение модели. Однако валидационный loss остаётся высоким, что подтверждает необходимость увеличения объема данных или применения дополнительных методов регуляризации.

2. Анализ предсказаний. Изучение примеров, где модель допустила ошибки, для понимания её слабых мест (рис. 2).

Тест 1:

- Длина: 500 нуклеотидов
- Предсказание: Мутация
- Вероятность: 0.5257
- Достоверность: 0.53

Рисунок 2 – Пример вывода

Модель демонстрирует способность к обучению, но страдает от переобучения из-за ограниченного размера датасета. Высокий recall при низком precision указывает на то, что модель склонна к предсказанию большинства примеров как мутаций. Это может быть связано с неоптимальным порогом классификации или недостаточной сложностью модели. Увеличение объема данных и применение методов регуляризации могут значительно улучшить качество модели.

Разработанная CNN показала потенциал для автоматизированного выявления патогенных мутаций в RB1, но требует доработок для повышения точности и устойчивости. Применение подобных моделей в клинической практике может ускорить анализ генетических данных и улучшить диагностику наследственных заболеваний, таких как ретинобластома.

Список использованных источников

1. Кнудсон, А. Г. Мутация и рак: статистическое исследование ретинобластомы / А. Г. Кнудсон // Труды Национальной академии наук США, 1971. – Т. 68, № 4. – С. 820–823.
2. ClinVar // Национальный центр биотехнологической информации США (NCBI). [Электронный ресурс]. – Режим доступа: URL: <https://www.ncbi.nlm.nih.gov/clinvar/>. – Дата доступа: 01.04.2025.
3. TensorFlow // Фреймворк для масштабируемого машинного обучения. [Электронный ресурс]. – Режим доступа: URL: <https://www.tensorflow.org/>. – Дата доступа: 27.03.2025.
4. BioPython // Библиотека для вычислительной молекулярной биологии. – Версия 1.81. – [Электронный ресурс]. – Режим доступа: URL: <https://biopython.org/>. – Дата доступа: 30.03.2025.
5. Педрегоса, Ф. Машинное обучение на Python: библиотека scikit-learn / F. Pedregosa, G. Varoquaux [и др.]; 2011. – Т. 12. – С. 2825–2830.

УДК 004.42+519.178

ВИЗУАЛИЗАЦИЯ ГРАФОВ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ C++ С ПОМОЩЬЮ УТИЛИТЫ Graphviz

Гречаников А. А., студ., Соколова А. С., ст. преп.
Витебский государственный технологический университет,
г. Витебск, Республика Беларусь

Реферат. В статье рассмотрена интеграция утилиты Graphviz в среду разработки Code::Blocks для визуализации графов на языке C++. Представлен практический пример использования алгоритма поиска минимального остовного дерева с автоматической генерацией графического представления.

Ключевые слова: Graphviz, C++, минимальное остовное дерево, алгоритм Прима, Code::Blocks.

В современном мире, где данные становятся все более сложными и взаимосвязанными, традиционные методы их анализа зачастую оказываются недостаточно эффективными. Социальные сети, биологические системы, IT-инфраструктуры, финансовые транзакции – все эти области объединяет одно: их суть можно раскрыть через графы – структуры, состоящие из узлов и связей. Однако работа с такими данными требует не только математических алгоритмов, но и инструментов, способных превратить абстрактные связи в наглядные визуальные модели.

Именно здесь на первый план выходят средства визуализации графов. Они не просто отображают информацию – они делают её понятной, превращая хаотичные массивы данных в карты отношений, где паттерны, аномалии и ключевые элементы становятся очевидными даже для неспециалистов. Будь то поиск мошеннических схем в банковских операциях, оптимизация логистических маршрутов или проектирование нейронных сетей, визуализация графов служит мостом между сложностью данных и человеческим восприятием.

Примеры инструментов:

- Gephi, Cytoscape – для анализа и визуализации сложных сетей.
- D3.js, Graphviz – библиотеки для программируемой визуализации.
- Neo4j (графовые базы данных) с встроенными инструментами отображения.

В данной работе был выбран для изучения инструмент Graphviz.

Graphviz – это набор инструментов с открытым исходным кодом для визуализации графов, разработанный AT&T Labs. Он позволяет преобразовывать текстовые описания графов в формате DOT в графические изображения (PNG, SVG, PDF и др.)

Этапы интеграции.

1. Установка Graphviz с добавлением в системный путь.
2. Настройка путей к библиотекам и заголовочным файлам в Code::Blocks.
3. Подключение необходимых библиотек: gvc.lib, cgraph.lib, cgraph++.lib, gvc++.lib gvc.h.

В качестве демонстрации работы Graphviz рассмотрим задачу нахождения минимального остовного дерева в графе. Минимальное остовное дерево (или минимальное покрывающее дерево) в (неориентированном) связном взвешенном графе – это остовное дерево этого графа, имеющее минимальный возможный вес, где под весом дерева понимается сумма весов входящих в него рёбер. Минимальное остовное дерево – это фундаментальная структура в теории графов, имеющая множество практических применений: от проектирования компьютерных сетей до анализа транспортных систем.

В разработанном примере пользователь вводит количество вершин графа, после чего вводит его матрицу смежности, а программа выдаёт готовый рисунок нашего графа с выделенным минимальным остовным деревом. На рисунке 1 представлен выбранный тестовый неориентированный граф с уже выделенным правильным ответом, а его матрица смежности – на рисунке 2.

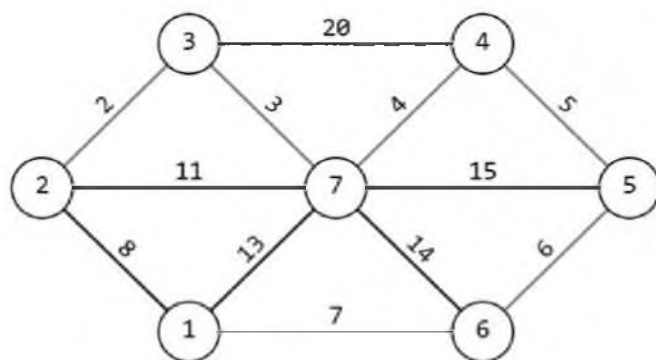


Рисунок 1 – Тестовый граф

Матрица смежности

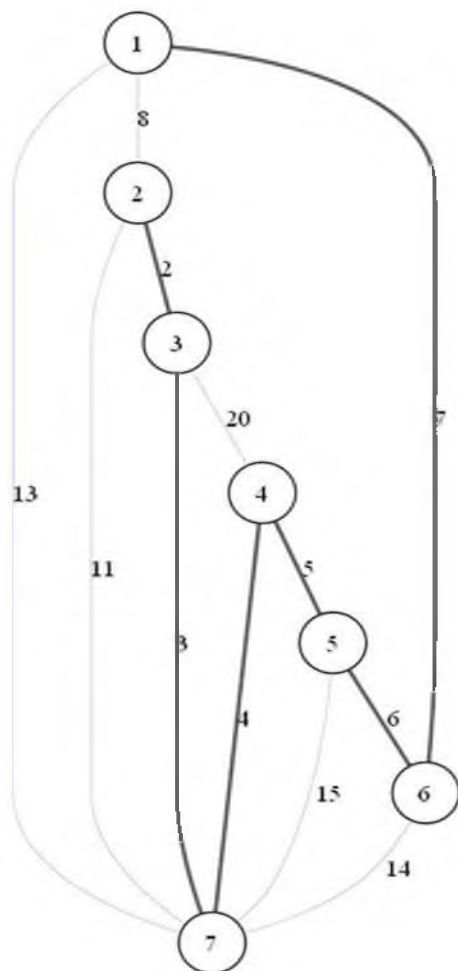
| | | | | | | |
|----|----|----|----|----|----|----|
| 0 | 8 | 0 | 0 | 0 | 7 | 13 |
| 8 | 0 | 2 | 0 | 0 | 0 | 11 |
| 0 | 2 | 0 | 20 | 0 | 0 | 3 |
| 0 | 0 | 20 | 0 | 5 | 0 | 4 |
| 0 | 0 | 0 | 5 | 0 | 6 | 15 |
| 7 | 0 | 0 | 0 | 6 | 0 | 14 |
| 13 | 11 | 3 | 4 | 15 | 14 | 0 |

Рисунок 2 – Матрица смежности тестового графа

Был написан код на языке C++, позволяющий найти минимальный остов, используя алгоритм Прима.

После получения данных программа получает решение и генерирует изображение с выделенным минимальным остовным деревом (рис. 3).

Разработанная программа показала возможность автоматической визуализации графов.



Сочетание C++ и Graphviz обеспечило высокую производительность и гибкость.

Подобным образом мы можем представить абстрактные данные в виде наглядных моделей, что ускоряет анализ, улучшает понимание систем, а также помогает находить неочевидные решения. Без таких инструментов работа с крупными или сложно связанными данными менее эффективна.

Список использованных источников

1. Graphviz Official Website [Электронный ресурс]. – Режим доступа: <https://graphviz.org/>. – Дата доступа: 03.04.2025.
2. Алгоритмы: построение и анализ / Кормен Т. [и др.]; пер. с англ. - 3-е изд. – М.: ООО «И. Д. Вильямс», 2013. – 1328 с.
3. Страуструп, Б. Язык программирования C++. Специальное издание. Пер. с англ. – М.: Издательский дом Бином, 2019. – 1136 с.

Рисунок 3 – Сгенерированное программой изображение

УДК 004.4

ИНФОРМАЦИОННАЯ СИСТЕМА УПРАВЛЕНИЯ ОБЩЕЖИТИЯМИ

Руммо Д. С., студ., Казаков В. Е., к.т.н., доц., Ринейский К. Н., начальник ЦИТ

*Витебский государственный технологический университет,
г. Витебск, Республика Беларусь*

Реферат. В статье представлен обзор разработки клиентской части информационной системы управления общежитиями, представлены предпосылки его внедрения в учреждение образования и круг решаемых им задач.

Ключевые слова: front-end, управление общежитиями, TypeScript, React.

Управление общежитиями – это информационная подсистема, предназначенная для автоматизации процессов учёта проживания студентов в общежитиях. Она регламентирует структуру и порядок взаимодействия с данными о комнатах, проживающих в них лицах, а также предоставляет механизмы заселения, переселения, выселения и подтверждения проживания. В системе реализован просмотр информации о комнатах и их текущем состоянии, оформление заявок на заселение или переселение, подтверждение факта заселения либо его отмена.