ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

**Отчет по лабораторной работе №3**

по курсу «Дискретная математика»

**«Алгоритм Шимбелла»**

Выполнил: студент группы РИС-23-3Б

Асташин Д. А.

Проверил: старший преподаватель кафедры ИТАС

Рустамханова Г.И.

2024 г.

**Цель работы**

Разработка программы, реализующей алгоритм Шимбелла для поиска минимальных и максимальных путей заданной длины между всеми парами вершин взвешенного ориентированного графа.

**Постановка задачи**

Нахождение кратчайших или самых длинных путей в заданное количество переходов с помощью алгоритма Шимбелла.

На вход подается: матрица расстояний взвешенного ориентированного графа, направление экстремума и количество переходов.

Вывод программы: матрица с исходами – путями минимальной или максимальной длины.

**Алгоритма Шимбелла**

Алгоритм Шимбелла основан на возведении матрицы смежности в степень по специальным правилам, отличным от стандартного перемножения матриц.

При возведении в степень k получается матрица, содержащая длины всех возможных путей, состоящих ровно из k рёбер, между каждой парой вершин графа.

Каждый элемент новой матрицы Ak[i][j] вычисляется по формуле:

Ak[i][j] = min(или max)(A[i][m] + A[m][j]) для всех m

Где A[i][m] и A[m][j] - длины рёбер между соответствующими вершинами. В отличие от обычного умножения матриц, здесь элементы складываются попарно, и из всех полученных значений выбирается минимальное (или максимальное, если ищем наибольший путь).

При этом, если одно из слагаемых равно нулю, это означает отсутствие ребра, и такое значение не учитывается при выборе минимума или максимума. Если для пары вершин не найдено ни одного допустимого пути, то результатом считается 0, что означает отсутствие пути заданной длины.

Таким образом, степень k отражает длину пути в рёбрах, а итоговая матрица показывает, какие минимальные или максимальные пути длины k существуют между всеми парами вершин.

**Реализация**

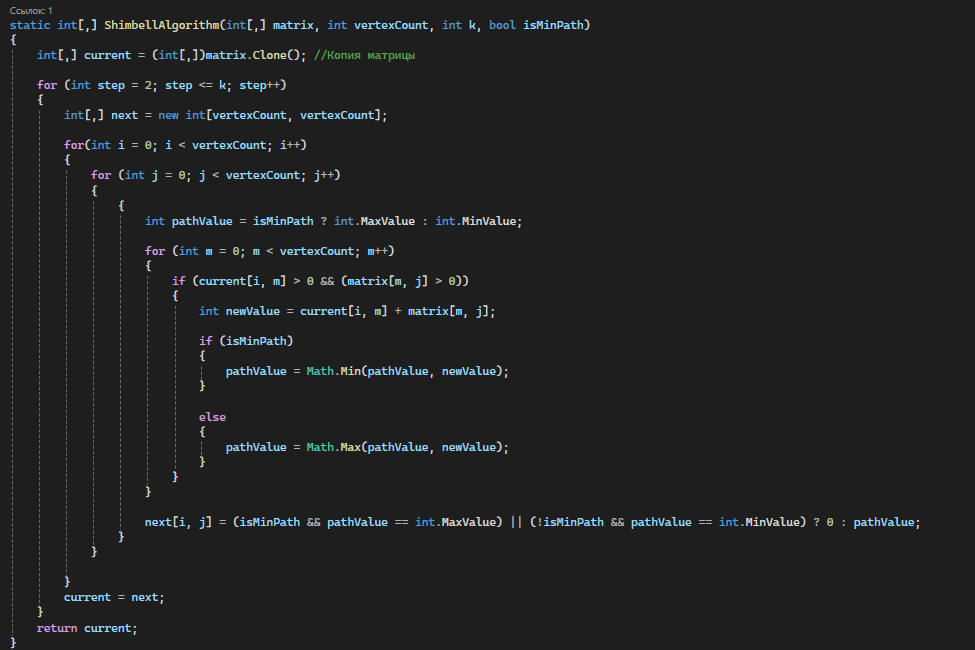
Алгоритм Шимбелла реализован в функции ShimbellAlgorithm (рисунок №1), которая принимает:

* Исходную матрицу смежности;
* Количество вершин;
* Значение k;
* Флаг isMinPath, определяющий тип вычисления (поиск минимального или максимального пути).

Алгоритм работает следующим образом:

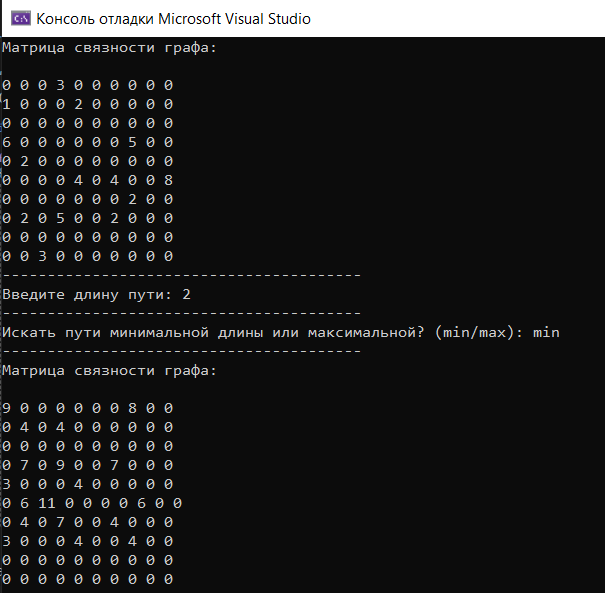
1. Копирование матрицы — создаётся рабочая копия исходной матрицы смежности;
2. Итерации по степени — начиная со второй степени, матрица последовательно возводится в степень до k включительно;
3. Обход пар вершин (i, j)— для каждой пары считается длина всех возможных путей длины k через промежуточную вершину m;
4. Условие наличия пути — учитываются только те случаи, где и A[i][m] и A[m][j] больше 0 (иначе путь невозможен);
5. Сравнение значений — найденные длины путей сравниваются и выбирается либо минимум, либо максимум, в зависимости от выбранного режима;
6. Отсутствие пути — если среди всех возможных путей не найден ни один (то есть длина так и осталась равна int.MaxValue или int.MinValue), то в итоговой матрице записывается 0.

После выполнения всех итераций возвращается матрица, в которой каждый элемент показывает минимальную или максимальную длину пути из вершины i в вершину j, состоящего ровно из k рёбер.



*Рис.1 – Реализация алгоритма Шимбелла*

**Результаты работы программы**

****

**Заключение**

В ходе выполнения работы был реализован алгоритм Шимбелла, позволяющий находить длины путей фиксированной длины между всеми парами вершин графа. Реализованная программа предоставляет возможность пользователю выбрать, искать ли минимальные или максимальные пути, а также задать желаемую длину пути (в количестве рёбер).

Программа корректно обрабатывает случаи отсутствия путей между вершинами, а также учитывает специфику работы с нулевыми значениями в матрице смежности. Алгоритм был протестирован на различных матрицах и показал правильные результаты как для поиска минимальных, так и максимальных путей.

[**GitHub**](https://github.com/astidii/PNRPU/tree/main/Discrete%20mathematics/Shimbell%20Algorithm)