

Документация

Введение

Эта документация описывает реализацию класса `multigraph` и различные алгоритмы, которые могут быть выполнены на нем. Код, приведенный в `main.cpp` файле демонстрирует использование класса `multigraph` и следующих алгоритмов:

- Depth-First Search (DFS) для нахождения компонентов и эйлеровых циклов в мультиграфе.
- Breadth-First Search (BFS) для поиска дополнительных путей в остаточной сети.
- Алгоритм сопоставления для нахождения дополняющих цепочек в сопоставлении.

Class Edge

Этот класс представляет ребро в мультиграфе. Он обладает следующими атрибутами:

- `source` -исходная вершина ребра.
- `destination` - конечная вершина ребра.
- `capacity` - пропускная способность ребра.

Class MultiGraph

Этот класс представляет собой мультиграф. Он имеет следующие приватные атрибуты:

- `numVertices` - количество вершин в мультиграфе.
- `adjacencyList` - представление списка смежности мультиграфа.

Класс предоставляет следующие публичные методы:

- `MultiGraph(int numVertices)` - конструктор для создания мультиграфа с заданным количеством вершин.
- `void addEdge(int source, int destination, int capacity)` - добавляет ребро к мультиграфу.
- `vector<int> findMaximumClique()` - находит максимальную клику в мультиграфе.
- `int findMaxFlow(int source, int sink, vector<int>& path)` - находит максимальный поток в сети, используя алгоритм Форда-Фалкерсона.
- `vector<int> findEulerianCycle()` - находит эйлеровский цикл в мультиграфе.
- `bool isEulerian()` - проверяет, имеет ли мультиграф цикл Эйлера.
- `vector<vector<int>> findStronglyConnectedComponents()` - находит сильно связанные компоненты в мультиграфе, используя алгоритм Косарайю.

Для получения подробных объяснений каждого метода, пожалуйста, обратитесь к комментариям к коду в `main.cpp`.