

1. Тема

Разработка программы решения теоретико-графовой задачи

2. Цель

При выполнении расчетной работы необходимо разработать и реализовать программу на C/C++, которая решает выданную преподавателем теоретико-графовую задачу.

3. Вариант и задачи

Вариант: определить числовую характеристику графа, а именно число компонентов связности неориентированного графа, с помощью матрицы инцидентности. Задачи для выполнения заданного варианта:

1. Разработать и реализовать алгоритм поиска в глубину для обхода графа.
2. Реализовать подсчет числа компонентов связности на основе алгоритма поиска в глубину.
3. Протестировать алгоритм на нескольких примерах графов.

4. Список ключевых понятий

- Граф: объект, состоящий из двух множеств: точек и линий, которые находятся между собой в некотором отношении.
- Неориентированный граф: граф, у которого множество линий, соединяющих любые пары вершин состоит только из линий, для которых несуществен порядок соединения вершин.
- Матрица инцидентности: представление графа в виде двумерного массива, где строки соответствуют вершинам, столбцы — рёбрам, а значение в ячейке указывает, инцидентна ли вершина ребру.
- Связной граф: граф, у которого у любых двух его вершин есть маршрут.
- Компонента связности: каждый из составляющих подграфов несвязного графа.
- Поиск в глубину (DFS): алгоритм нахождения решения, когда в каждой исследуемой вершине дерева решений выбирается один из возможных путей и исследуется до конца; при этом другие пути не рассматриваются, пока сохраняется возможность получить конкретный результат. Если такая возможность отсутствует, процесс поиска продолжается от ближайшей вершины дерева решений, имеющей неисследованные пути.

5. Тестовые примеры

Пример 1:

Входная матрица инцидентности:

```
1 0 0
1 1 0
0 1 1
0 0 1
```

Выходное значение:

Количество компонентов связности матрицы 1: 1

Пример 2:

Входная матрица инцидентности:

```
0 0 0
1 0 0
0 1 1
0 1 0
0 0 1
```

Выходное значение:

Количество компонентов связности матрицы 2: 3

6. Детализация преобразования входной конструкции в выходную

Пример преобразования на основе входной матрицы:

Входная матрица инцидентности:

```
1 0 0
1 1 0
0 1 1
0 0 1
```

Шаг 1. Инициализация:

- Количество вершин: 4.
- Вектор просмотренных вершин: `viewed = [false, false, false, false]`.
- Число компонентов связности: 0.

Шаг 2. Первый вызов функции DFS для вершины 0:

- Начинаем с вершины 0, отмечаем её как просмотренную: $\text{viewed} = [\text{true}, \text{false}, \text{false}, \text{false}]$.
- Обход соседей вершины 0:
 1. Находим вершину 1, которая инцидентна тому же ребру.
 2. Отмечаем вершину 1 как просмотренную: $\text{viewed} = [\text{true}, \text{true}, \text{false}, \text{false}]$.
- Обход соседей вершины 1:
 1. Находим вершину 2, которая инцидентна тому же ребру.
 2. Отмечаем вершину 2 как просмотренную: $\text{viewed} = [\text{true}, \text{true}, \text{true}, \text{false}]$.
- Обход соседей вершины 2:
 1. Находим вершину 3, которая инцидентна тому же ребру.
 2. Отмечаем вершину 3 как просмотренную: $\text{viewed} = [\text{true}, \text{true}, \text{true}, \text{true}]$.
- Обход соседей вершины 3:
 1. Не находим вершину, которая инцидентна тому же ребру.
 2. Завершаем функцию DFS.
- Число компонент связности: 1.

Шаг 3. Проверка оставшихся вершин:

- Вершины 1, 2 и 3 тоже просмотрены: $\text{viewed} = [\text{true}, \text{true}, \text{true}, \text{true}]$.
- Обход завершается.

Результат:

Количество компонент связности матрицы 1: 1

7. Вывод

В ходе расчетной работы был разработан алгоритм подсчёта числа компонент связности в неориентированном графе с использованием поиска в глубину. Алгоритм успешно реализован на языке C++, протестирован на двух примерах графов.

8. Список использованных источников

- Гладков, Л. А. Дискретная математика : учебное пособие / Л. А. Гладков, В. В. Курейчик, В. М. Курейчик ; под редакцией В. М. Курейчика. — Москва : ФИЗМАТЛИТ, 2014.