Санкт-Петербургский Политехнический Университет Петра Великого

Физико-механический институт

Кафедра Прикладной математики и Информатики

**Отчет по Лабораторной работе №3**

Тема: Проверка свойства древочисленности (ордерево)

Предмет: Дискретная математика

Студент: Егоркин Станислав Дмитриевич

Группа: 5030102/20202

Преподаватель: Нахатович Михаил Алексеевич

1. **Требования:**

Проверить, является ли орграф ордеревом по определению. Если нет, то в выходной файл написать, что нарушено. По итогу нужно запустить программу на различных орграфах и увидеть, что либо орграф является ордеревом и древочисленный, либо то, что если хотя бы одно из условий из определения ордерева не выполнено, то орграф как может, так и не может быть древочисленным.

Программа реализована на языке C++.

1. **Идея алгоритма:**

Орграф является ордеревом при соблюдении всех следующих условиях:

1. Наличие корня
2. Полустепень захода всех вершин (кроме корня) равна 1
3. Каждая вершина (кроме корня) достижима из корня.

Если хотя бы одно из условий не выполнено – орграф не является ордеревом.

Орграф является древочисленным если q(G)=p(G)-1, где q – количество рёбер, p – количество вершин графа.

1. **Алгоритмы программы:**

Поиск в ширину (BFS) - фундаментальный алгоритм обхода графа. Он начинается с узла, затем сначала обходит все смежные с ним узлы. После обхода всех смежных узлов обходятся их смежные узлы.

Алгоритм **BFS**:

1. **Инициализация**: заданная вершина помещается в очередь и помечается как «посещенная».
2. **Исследование:**пока очередь не пуста извлекается узел из очереди и обрабатывается. Каждый «непосещенного» соседний узел ставится в очередь и отмечается «посещенный»
3. **Завершение:**Повторяется шаг 2, пока очередь не опустеет.
4. **Описание работы кода:**
5. Нахождение корня (hasRoot):

Находим каждую полустепень захода. Если у вершины степень захода 0 – вершина является корнем.

1. Проверка полустепеней захода (checkInDegree):

Проверяем каждую полустепень захода в вершины. Если полустепень для каждой вершины (кроме корня) не равно 1, то выводится ошибка – орграф не ордерево.

1. Проверка достигаемости из корня (checkReachability):

Используется алгоритм BFS начинающийся с корня и проверяется, что каждая вершина была в очереди. Если есть хотя бы 1 вершина, которая не была в очереди, то до нее нельзя дойти из корня – орграф не ордерево.

5. **Пример работы алгоритма:**

Рассмотрим пример орграфа, записанный в виде ребер орграфа:

Изображение выглядит как круг, зарисовка, графическая вставка

Автоматически созданное описание

3 2

0 1

0 2

* + - 1. Находим корень: степень захода в вершину 0 равна 0, значит 0 – корень.
      2. Проверяем полустепени захода: 0 – 0, 1 – 1, 2 – 1. Для каждой вершины, кроме корня, полустепень захода равна 1, условие не нарушено.
      3. Проверка достижимости с помощь алгоритма BFS: берется вершина 0. Она ставится в очередь и отмечается «посещенной». В очередь ставится вершина 1 и отмечается «посещенной». Далее вершина 2. Все вершины отмечены «посещенными», значит из корня можно достичь каждую вершину.
      4. Проверяем, является ли орграф древочисленным. Вершин – 3. Ребер – 2. 2=3-1 (верно). Является древочисленным.

1. **Тесты:**
2. Граф в алгоритме:

Древочисленный граф.

Корень найден в вершине: 0

Ордерево.

1. Граф с несколькими корнями:

4 2

0 1

2 3

**Изображение выглядит как круг, зарисовка, белый

Автоматически созданное описание**

Не древочисленный граф.

Корень должен быть один.

0 - 0

1 - 1

2 - 0

3 - 1

Не ордерево.

1. Отсутствие полустепени захода:

4 4

0 1

0 2

1 3

2 3

Изображение выглядит как круг, диаграмма

Автоматически созданное описание

Не древочисленный граф.

Корень найден в вершине: 0

Нарушение степени полузахода

3 - 2

Не ордерево.

.

1. Отсутствие корня:

4 4

0 1

0 2

1 3

3 0

Изображение выглядит как круг

Автоматически созданное описание

Не древочисленный граф.

Кореня не найдено.

0 - 1

1 - 1

2 - 1

3 - 1

Не ордерево.

1. Вершины недостижимы до корня:  
   4 3

0 1

1 2

2 0

Изображение выглядит как круг

Автоматически созданное описание

Древочисленный граф.

Корень найден в вершине: 3

0 - недостижима из корня

1 - недостижима из корня

2 - недостижима из корня

Не ордерево.

1. Нарушение степени полузахода и достижимости:

5 5

0 1

1 2

2 0

1 3

3 2

Изображение выглядит как круг, диаграмма, линия

Автоматически созданное описание

Не древочисленный граф.

Корень найден в вершине: 4

Нарушение степени полузахода

2 - 2

0 - недостижима из корня

1 - недостижима из корня

2 - недостижима из корня

3 - недостижима из корня

Не ордерево.

1. Не ордерево, но является граф древочисленный:

Изображение выглядит как круг, диаграмма

Автоматически созданное описание

4 3

0 1

0 2

3 1

Древочисленный граф.

Корень должен быть один.

0 - 0

1 - 2

2 - 1

3 - 0

Не ордерево.

Приведены примеры (ордерево – 1, несколько корней – 2, нарушение полустепени захода – 3, отсутствие корня – 4, нарушение достижимости – 5, нарушение полустепени захода и достижимости – 6, не ордерево, но древочисленный граф – 7)

1. **Формат данных**

Данные указываются в виде направленных рёбер, формата число-пропуск-число, где первое число является начальным, а второе конечным. Числа должны быть натуральными от 0 до (количество вершин-1).

Использование записи орграфа в виде рёбер обусловлено для легкой проверки на древочисленность графа.

В ходе работы алгоритма орграф записывается в виде вектор векторов, где векторы соответствуют вершинам и их количество равно количеству вершин, а вектор в векторе соответствует всем соединенным вершинам и его размер равен количеству соединенных вершин. Данное отображение данных нужно для проверки достигаемости из корня, а именно алгоритма BFS.

1. **Область применимости**

Количество вершин и ребер задаётся в формате неотрицательных целочисленных чисел в пределах до 2147483647. Далее, через пробел, 2 числа, описывающие название вершин, которые показываются, как в орграфе из одной вершины следует ребро связности к другой вершине.

1. **Представления графа**

Орграф задается в виде рёбер связности, где первое число – вершина, от которой идёт ребро связности ко второму числу (второй вершины)

1. **Вывод**

Данный алгоритм позволяет проверить орграф по свойствам. Сложность алгоритма можно найти из каждой проверки :

1. Поиск корня – О(n)
2. Полустепень захода – O(n)
3. Проверка достижимости – равна сложности BFS – O(n+m), где n и m – количество ребер и вершин.

Сложность алгоритма – O(n+m).

Источники информации:

1. https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/