Санкт-Петербургский Политехнический Университет Петра Великого

Физико-механический институт

Кафедра Прикладной математики и Информатики

**Отчет по Лабораторной работе №2**

Тема: Проверка изоморфизма графов

Предмет: Дискретная математика

Студент: Егоркин Станислав Дмитриевич

Группа: 5030102/20202

Преподаватель: Нахатович Михаил Алексеевич

1. **Требования:**

Требуется реализовать алгоритм проверки изоморфизма двух графов. Придумать алгоритм самостоятельно, либо найти алгоритм в какой-нибудь статье. Если будет реализована самая очевидная проверка изоморфизма, то в отчёте нужно отразить то, как можно было бы улучшить алгоритм.

Два графа, которые передаются на вход программе, могут быть описаны как в одном файле, так и в двух – на выбор студента. Если два графа изоморфны, то результатом работы алгоритма должен быть изоморфизм.

Программа реализована на языке C++

1. **Идея алгоритма:**

Графы изоморфны тогда, когда у них совпадает матрица смежности при возможных перестановках вершин (или отсутствия перестановок)

1. **Алгоритмы программы:**

1) Проверка количества вершин - если количество вершин в графах отличается, они точно не изоморфны.

2) Сравнение матрицы смежности - если матрицы одинаковые, то графы – изоморфны.

3) Перебор возможных перестановок вершин:

Перебрать все возможные перестановки вершин одного графа и проверять, совпадает ли его матрица смежности с матрицей другого графа.

1. **Описание работы:**
2. Основная функция (checkIsomorphism):

В данной функции перебираются все возможные перестановки вершин, до тех пор, пока не будет получены одинаковые матрицы смежности либо не перебраны все возможные перестановки.

1. Проверка (areIsomorphic):

Проверяет, совпадают ли матрицы смежности обоих графов.

1. **Пример работы алгоритма:**

Рассмотрим графы, заданные в форме матрицы смежности:

Изображение выглядит как диаграмма, зарисовка, линия, белый

Автоматически созданное описание

1. Проверка количества вершин – верно
2. Проверка начальной матрицы смежности - отличаются
3. Перебор: чтобы изменить 2 вершины (i,k), необходимо в матрице смежности изменить i-ую и k-ую строчку и столбец.
4. Меняем в первой матрице 1 вершину со 2. Матрицы смежности в данном варианте выглядят так:

Матрицы не одинаковы.

1. Меняем в первой матрице 2 вершину с 3. Матрицы смежности : (как меняются строки и столбцы)
   1. Меняем 2 и 3 столбец.
   2. Меняем 2 и 3 строку.

Получаем :

Матрицы одинаковы, значит графы изоморфны.

1. **Тесты:**
2. Различие вершин – графы не изоморфны.
3. Пример аналогичный отчету – графы изоморфны.
4. Пример с 4 вершинами – графы вида:

Изображение выглядит как диаграмма, линия, круг

Автоматически созданное описание

Вывод – графы изоморфны.

1. **Улучшение:**

Существует несколько возможных улучшений для уменьшения времени работы простого перебора:

1. Использование инвариантов графа: можно сравнивать более сложные инварианты вершин, такие как спектр матрицы смежности или количество треугольников вокруг вершины. Это позволит отсечь некоторые варианты перестановок раньше.
2. Поиск канонического представления графа: вместо полного перебора всех перестановок вершин можно искать "каноническое" представление графа, что может ускорить процесс проверки.

3) Алгоритм *nauty*

Алгоритм *nauty* гораздо более эффективен, чем простой перебор, так как использует специальные структуры данных для поиска канонического представления графа и сравнения графов на изоморфизм. Основные идеи:

1. Классы эквивалентности вершин: Алгоритм разделяет вершины графа на классы эквивалентности по их инвариантам. Это уменьшает количество возможных перестановок.
2. Поиск канонической метки графа: В процессе алгоритма строится каноническая метка (label) для каждого графа, которая позволяет быстро сравнивать графы.
3. **Область применимости**

Количество вершин задаётся в формате неотрицательных целочисленных чисел в пределах до 2147483647, матрицы смежности задаются в отдельных файлах (для удобства пользования) в формате квадратной матрицы размером количества вершин и заполняются данными 0 и 1.

1. **Область применимости**

Количество вершин задаётся в формате неотрицательных целочисленных чисел в пределах до 2147483647, матрица смежности в формате квадратной матрицы размером количества вершин и заполняются данными 0 и 1.

1. **Вывод**

Данный алгоритм очень простой и рабочий, но работает за факториальное время O(n!), что делает его непрактичным для больших графов.

Источники информации:

1. <https://ru.hexlet.io/courses/graphs/lessons/isomorphism/theory_unit>
2. https://ru.wikipedia.org/wiki/Изоморфизм\_графов