

Лабораторная работа № 9 по курсу дискретного анализа: Графы

Выполнил студент группы 08-308 МАИ *Попов Николай*.

Условие

Задан неориентированный граф, состоящий из n вершин и m ребер. Вершины пронумерованы целыми числами от 1 до n . Необходимо вывести все компоненты связности данного графа.

Метод решения

Понятие компоненты связности вытекает из понятия связности графа. Компонента связности - набор вершин графа, между любой парой которых существует путь. Общее понятие связности распространяется только на неориентированные графы. Для описания ориентированных графов используются понятия сильной и слабой связности. Для поиска компонент связности используется обычный DFS. При запуске обхода из одной вершины, он гарантированно посетит все вершины, до которых возможно добраться, то есть, всю компоненту связности, к которой принадлежит начальная вершина. Для нахождения всех компонент просто попытаемся запустить обход из каждой вершины по очереди, если мы ещё не обошли её компоненту ранее.

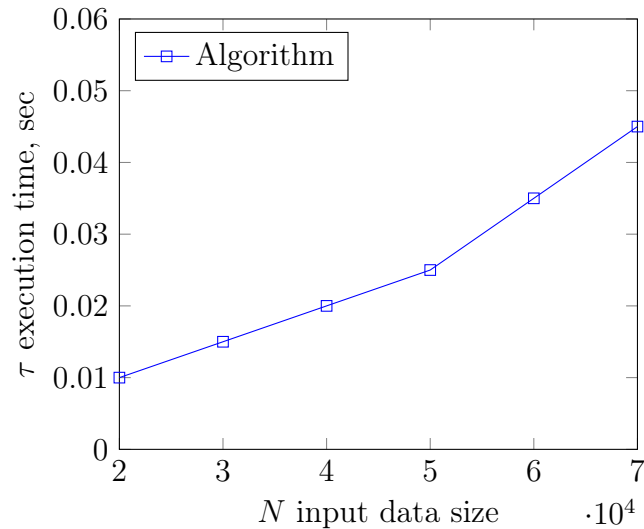
Описание программы

Основные этапы работы программы:

1. Считываем входные данные и строим список смежностей для представления графа.
2. Учитывая, что в худшем случае может быть n компонент связности (если ни одна вершина не связана с другой), запускаем алгоритм поиска в глубину (DFS) n раз.
 - Изначально массив *check*, показывающий, какие вершины уже были рассмотрены, заполнен значением -1.
 - После выполнения поиска в глубину все вершины, находящиеся в одной компоненте связности, в массиве *check* будут иметь одинаковое значение, соответствующее номеру этой компоненты связности.
3. После завершения поиска компонент связности, записываем в массив *result* вершины, входящие в каждую компоненту связности.

Тест производительности

Зависимость времени выполнения от размера входных данных



По результатам тестирования видно, что сложность выполнения алгоритма является линейной. Так как мы обходим только непересекающиеся компоненты связности, а значит по итогу выполнения алгоритма пройдем весь граф только 1 раз, не заходя в одну и ту же вершину несколько раз. Мы знаем, что сложность обхода графа составляет $O(n + m)$, где n — количество вершин, а m — рёбер графа, значит, сложность алгоритма поиска компонентов связности тоже $O(n + m)$.

Выводы

В результате проведенной лабораторной работы мной была решена задача по поиску компонент связности графа.