

Лабораторная работа № 9 по курсу Дискретный Анализ. Графы

Выполнил студент группы 08-307 МАИ *Павлов Иван*.

Условие

Кратко описывается задача:

Вариант 3. Поиск компонент связности

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.

Описание алгоритма

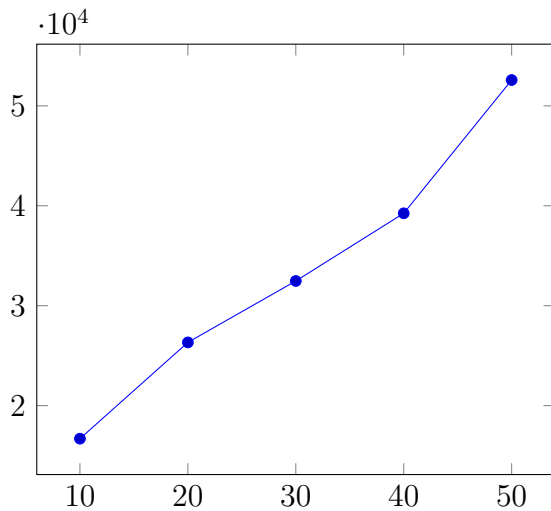
Чтобы найти все компоненты связности в неориентированном невзвешенном графе, необходимо провести серию обходов в глубину. При обходе в глубину из первой вершины все достижимые из нее вершины станут помеченными, далее запускаем dfs только из еще не помеченных вершин. На каждом шаге запоминаем все помеченные вершины - это и есть компоненты связности. Также я сортировал на каждом шаге все помеченные вершины и в конце сортировал сам граф для выполнения условия.

Описание программы

Программа хранит граф в виде списка смежности, dfs реализован рекурсивно, сортировку использовал стандартную.

Тест производительности

Оценим сложность данного алгоритма, проведя тест на производительность. Ниже представлен график зависимости времени работы от количества вершин и ребер ($n = m$) Время беру в наносекундах.



В общем случае этот алгоритм имеет сложность $O(n + m)$, так как сложность dfs составляет $O(n + m)$ (каждая вершина и ребро посещается не более 1 раза), и в серии обхода мы посещаем каждую вершину не более одного раза, сохраняя сложность. Также на каждом шаге выполняется сортировка компонент, что дает сложность $O(n \log n)$, и внешняя сортировка всех компонент также за $O(n \log n)$. Таким образом алгоритм имеет общую сложность $O(n \log n + m)$.

Выводы

Поиск в глубину оказался эффективным инструментом для исследования графов, особенно в задаче нахождения компонент связности. Его способность исследовать граф до самого "глубокого" уровня перед возвратом к другим ветвям делает его идеальным для этой цели. Алгоритм масштабируется линейно с увеличением числа рёбер и вершин, хотя сортировка компонент связности добавляет логарифмический фактор. Это делает алгоритм приемлемым для использования на средних и даже на больших графах.