

Лабораторная работа № 9 по курсу дискретного анализа: Графы

Выполнил студент группы М8О-308Б-20 МАИ *Зинин Владислав*.

Условие

1. Разработать программу на языке C или C++, реализующую указанный алгоритм. Формат входных и выходных данных описан в варианте задания. Первый тест в проверяющей системе совпадает с примером.
2. **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.

Метод решения

Поскольку заданный граф не является ориентированным, можно использовать обход в глубину для решения данной задачи. При запуске обхода из вершины, принадлежащей к некоторой компоненте связности, обход посетит все вершины из этой компоненты и только их. Таким образом, в функцию обхода можно передать вектор, в который будут помещаться вершины из очередной компоненты связности. Считываем заданный неориентированный граф, запускаем обходы в глубину из всех его вершин, сохраняя при этом результат в вектор, передаваемый по ссылке. По итогу наша сложность совпадает со сложностью обхода в глубину, то есть $O(V + E)$, где V и E — общее количество вершин и ребер в графе соответственно.

Описание программы

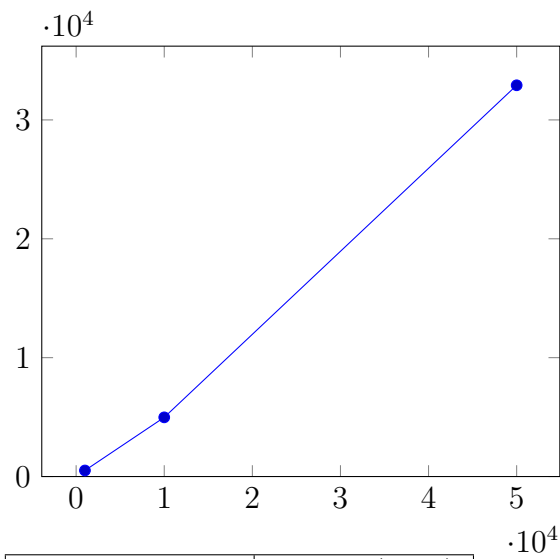
Программа состоит из одного файла.

Дневник отладки

Программа была выполнена с первого раза.

Тест производительности

Убедимся, что данный алгоритм действительно имеет линейную сложность. Для этого замерим время программы для графов с следующим количеством вершин: 1000, 10000 и 50000.



Кол-во вершин	Время (в мс)
1000	510
10000	4980
50000	32914

Таким образом, мы видим, что сложность данного алгоритма - линейная.

Выводы

В результате проведенной лабораторной работы я повторил написание обхода неориентированного графа в глубину и научился его модифицировать для поиска компонент связности.