Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №9 по курсу «Дискретный анализ»

Студент:	Ю. Ю. Обыденкова
Преподаватель:	А. Н. Ридли
Группа:	М8О-308Б-18
Дата:	
Оценка:	
Подпись:	

Москва, 2021

Лабораторная работа №9

Задача: Разработать программу на языке C или C++, реализующую указанный алгоритм согласно заданию:
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.

Формат входных данных: В первой строке заданы $1 \leq n \leq 10^5$ и $1 \leq m \leq 10^5$. В следующих m строках записаны ребра. Каждая строка содержит пару чисел – номера вершин, соединенных ребром.

1 Описание

В связи с тем, что граф неориентированный, 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 передавать вектор, в который будут помещаться вершины из очередной компоненты связности. Сложность совпадает со сложностью обхода в глубину, то есть $O(V + E)$.

2 Исходный код

После считывания неориентированного графа запускаются обходы в глубину из всех его вершин, результат сохраняется в вектор, передаваемый по ссылке.

```
#include <iostream>
#include <vector>

#define NON_SET_COMPONENT -1


using Graph = std::vector<std::vector<int>>;

void dfs(int u, const Graph& g, std::vector<int>& components, int cur_comp) {
    components[u] = cur_comp;
    for (int v : g[u]) {
        if (components[v] == NON_SET_COMPONENT) {
            dfs(v, g, components, cur_comp);
        }
    }
}

int main() {
    int n,m;
    std::cin >> n >> m;
    Graph g(n);
    std::vector<int> comp_number(n,NON_SET_COMPONENT);
    for (int i = 0; i < m; ++i) {
        int a, b;
        std::cin >> a >> b;
        a--;
        b--;
        g[b].push_back(a);
        g[a].push_back(b);
    }

    int cur_comp = 0;
    for (int i = 0; i < n; ++i) {
        if (comp_number[i] == NON_SET_COMPONENT) {
            dfs(i, g, comp_number, cur_comp);
            cur_comp++;
        }
    }

    std::vector<std::vector<int>> result(cur_comp);
    for (int i = 0; i < comp_number.size(); ++i) {
        result[comp_number[i]].push_back(i);
    }
    for (int i = 0; i < result.size(); ++i) {
```

```
        for (int v : result[i]) {
            std::cout << v + 1 << " ";
        }
        std::cout << "\n";
    }

    return 0;
}
```

3 Консоль

```
ulia@WIN-9LNCMF0CCPQ:~$ g++ lab9.cpp -o lab9
ulia@WIN-9LNCMF0CCPQ:~$ ./lab9
5 4
1 2
2 3
1 3
4 5

1 2 3
4 5
ulia@WIN-9LNCMF0CCPQ:~$
```

4 Выводы

Выполнив девятую лабораторную работу по курсу «Дискретный анализ», я больше узнала о графах и алгоритмах работы с ними. Теория графов к настоящему времени содержит достаточно много эффективных инструментов для решения столь же широкого круга проблем. Однако, средства и идеи, сегодня относящиеся к области дискретной математики, именуемой теорией графов, пребывают по сей момент в некотором единстве.

Так, в решаемой мной задаче , можно заметить, что граф неориентированный, и можно использовать обход в глубину. 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 передавать вектор, в который будут помещаться вершины из очередной компоненты связности.
Сложность совпадает со сложностью обхода в глубину, то есть $\mathrm{O(V + E)}$.

Существуют и другие алгоритмы поиска кратчайшего пути от одной вершины графа до другой. Например, алгоритм Дейкстры. Этот алгоритм является жадным и имеет сложность $O(VlogV)$, когда алгоритм Форда-Беллмана имеет сложность $O(V * E)$. Однако, первый не умеет работать с ребрами, имеющими отрицательный вес, поэтому для моей задачи он не применим.

Список литературы

[1] Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. *Алгоритмы: построение и анализ, 2-е издание.* — Издательский дом «Вильямс», 2007. Перевод с английского: И. В. Красиков, Н. А. Орехова, В. Н. Романов. — 1296 с. (ISBN 5-8459-0857-4 (рус.))