

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования

**АДЫГЕЙСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ**

Инженерно-физический факультет

Кафедра автоматизированных систем обработки информации
и управления

Отчет по практике

Обход графа в глубину и ширину.

2 курс, группа ИВТ АСОИУ

Выполнил:

_____ В. А. Сапунов

«05» 06. 2025 г.

Руководитель:

_____ С. В. Теплоухов

«05» 06. 2025 г.

Майкоп, 2025 г.

1. Введение

- 1) Вариант 8 - Обход графа в глубину и ширину.
- 2) Пример кода, решающего данную задачу
- 3) Скриншот программы
- 4) Литература и источники

2. Ход работы

2.1 Теория решения

Формальное определение графа

Граф $G = (V, E)$, где:

$V = \{v_1, v_2, \dots, v_n\}$ - множество вершин

$E \subseteq V \times V$ - множество рёбер

Матрица смежности

Для графа с n вершинами матрица смежности A размером $n \times n$, где:

$A[i][j] = 1$, если существует ребро из v_i в v_j

0, иначе

Обход в ширину (BFS)

Формально можно описать как последовательность множеств:

$L = \{s\}$ (начальная вершина)

$L = \{v \in V \mid \exists u \in L: (u,v) \in E\} \setminus \bigcup_{k=0}^{i-1} L_k$

где L_i - множество вершин на расстоянии i от начальной вершины s .

Обход в глубину (DFS)

2

Рекурсивное определение:

DFS(v):

посещаем v

для всех $u \in \text{Adj}(v)$:

если u не посещена:

DFS(u)

где $\text{Adj}(v)$ - множество вершин, смежных с v

2.2 Код приложения

```
#include <iostream>
```

```
#include <vector>
```

```
#include <queue>
```

```
#include <stack>
```

```
using namespace std;
```

```
void BFS(const vector<vector<int>>& graph, int startNode) {
```

```

int n = graph.size();
vector<bool> visited(n, false);
queue<int> q;

q.push(startNode);
visited[startNode] = true;

cout << "Обход BFS, начиная с узла " << startNode << ": ";

while (!q.empty()) {
    int currentNode = q.front();
    q.pop();
    cout << currentNode << " ";

    for (int neighbor : graph[currentNode]) {
        if (!visited[neighbor]) {
            visited[neighbor] = true;
            q.push(neighbor);
        }
    }
}
cout << endl;
}

void DFS_recursive(const vector<vector<int>>& graph, vector<bool>& visited, int currentNode) {
    visited[currentNode] = true;
    cout << currentNode << " ";

    for (int neighbor : graph[currentNode]) {
        if (!visited[neighbor]) {
            DFS_recursive(graph, visited, neighbor);
        }
    }
}

void DFS_iterative(const vector<vector<int>>& graph, int startNode) {
    int n = graph.size();
    vector<bool> visited(n, false);
    stack<int> s;

    s.push(startNode);
    visited[startNode] = true;

    cout << "DFS (итеративный) обход, начинающийся с узла " << startNode << ": ";

    while (!s.empty()) {

```

```

        int currentNode = s.top();
        s.pop();
        cout << currentNode << " ";

        for (auto it = graph[currentNode].rbegin(); it != graph[currentNode].rend(); ++it)
            if (!visited[*it]) {
                visited[*it] = true;
                s.push(*it);
            }
    }
    cout << endl;
}

int main() {
    setlocale(0, "ru");
    vector<vector<int>> graph = {
        {1, 2},
        {0, 3, 4},
        {0, 5},
        {1},
        {1},
        {2}
    };

    BFS(graph, 0);

    vector<bool> visited(graph.size(), false);
    cout << "Обход DFS (рекурсивный), начинающийся с узла 0: ";
    DFS_recursive(graph, visited, 0);
    cout << endl;

    DFS_iterative(graph, 0);
}

```

3. Изображение кода программы

Скриншот программы представлен на рисунке 1

```

Консоль отладки Microsoft Vi
Обход BFS, начиная с узла 0: 0 1 2 3 4 5
Обход DFS (рекурсивный), начинающийся с узла 0: 0 1 3 4 2 5
DFS (итеративный) обход, начинающийся с узла 0: 0 1 3 4 2 5

C:\Users\wowa2\OneDrive\Рабочий стол\c++\ConsoleApplication2\x64
работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно:

```

4. Литература и источники

Основные алгоритмы: Кормен, Т., Лейзерсон, Ч., Ривест, Р., Штайн, К. "Алгоритмы: построение и анализ". 3-е изд. - М.: Вильямс, 2022. (Глава 22 "Элементарные алгоритмы для работы с графами") Скиена, С. "Алгоритмы. Руководство по разработке". 2-е изд. - СПб.: БХВ-Петербург, 2011.

Теоретические основы: Diestel, R. "Graph Theory". 5th edition. Springer, 2017. Harary, F. "Graph Theory". Addison-Wesley, 1969.

Оригинальные работы: BFS: Описан в работе Moore E.F. "The shortest path through a maze". Proceedings of the International Symposium on the Theory of Switching (1959) DFS: Впервые формально описан Тарьяном в Tarjan R.E. "Depth-first search and linear graph algorithms". SIAM Journal on Computing, 1972

5. Репозиторий: <https://github.com/SapunovVladimir/Pracitce2025>