Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа № 6

По дисциплине ЯПИС

За шестой семестр

## Тема: «Метод поиска в глубину на графе. Рекурсивная и не рекурсивная реализации алгоритма DFS»

Вариант 8

Выполнил:

Студент 3 курса

Группы ИИ-16 (2)

Пилипук З.А.

Проверил:

Слинко Е.В.

Брест, 2021

**Цель:** Сформировать представление о методе и алгоритме поиска в глубину на графе и его применении при решении задач (задачи класса сложности Р).

**Ход работы**

**Вариант 7.** Разработайте и напишите программу нахождения количества связных компонент графа, используя поиск в глубину для определения всех вершин, принадлежащих одной связанной компоненте (определяется путь, связывающий между собой любую пару вершин). Рассмотрите случай, когда граф оказывается несвязным.

**Код программы:**

#include <iostream>

#include <vector>

using namespace std;

int n = 3;

vector<vector<int>> g;

bool used[3];

vector<int> comp;

void dfs(int v) {

used[v] = true;

comp.push\_back(v);

for (size\_t i = 0; i < g[v].size(); ++i) {

int to = g[v][i];

if (!used[to])

dfs(to);

}

}

void find\_comps() {

for (int i = 0; i < n; ++i)

used[i] = false;

for (int i = 0; i < n; ++i)

if (!used[i]) {

comp.clear();

dfs(i);

cout << "Component:";

for (size\_t j = 0; j < comp.size(); ++j)

cout << ' ' << comp[j];

cout << endl;

}

}

int main()

{

g.push\_back({ 1,2,3 });

g.push\_back({ 0,2,3 });

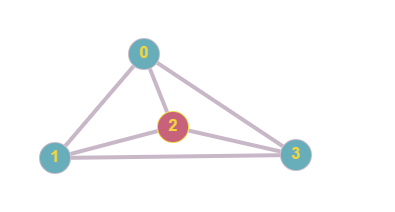
g.push\_back({ 0,1,3 });

g.push\_back({ 0,1,2 });

find\_comps();

return 0;

}





**Вывод:** Сформировали представление о методе и алгоритме поиска в глубину на графе и его применении при решении задач.