Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный технический университет”

Кафедра ИИТ

Лабораторная работа №6

По дисциплине ЯПИС

Тема: «Метод поиска в глубину на графе»

**Выполнил:**

Студент 3 курса

группы ИИ-16

Пастушик М.С.

**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).

**Задание**

5. Напишите программу нахождения простого пути в графе из вершины v в w с использованием процесса поиска в глубину. Программа распечатывает простой путь (в виде последовательности вершин найденного пути) из заданной вершины v к искомой вершине. Используйте представление графа в виде списка смежных вершин.

**Текст программы:**

#include <iostream>

#include <map>

#include <vector>

#include <stack>

class Graph

{

public:

std::map<int, bool> visited;

std::map<int, std::vector<int>> adj;

void addEdge(int v, int w);

void DFS(int v, int end);

private:

bool exit = false;

};

void Graph::addEdge(int v, int w)

{

adj[v].push\_back(w);

adj[w].push\_back(v);

}

void Graph::DFS(int v, int end)

{

visited[v] = true;

std::cout << v << " ";

if(v == end) {

exit = true;

return;

}

std::vector<int>::iterator i;

for (i = adj[v].begin(); i != adj[v].end(); ++i) {

if (!visited[\*i])

DFS(\*i, end);

if(exit) return;

}

}

int main()

{

Graph g;

g.addEdge(1, 2);

g.addEdge(1, 3);

g.addEdge(1, 0);

g.addEdge(3, 4);

g.addEdge(2, 5);

g.addEdge(8, 5);

g.addEdge(8, 6);

g.addEdge(8, 7);

g.addEdge(8, 9);

g.addEdge(0, 9);

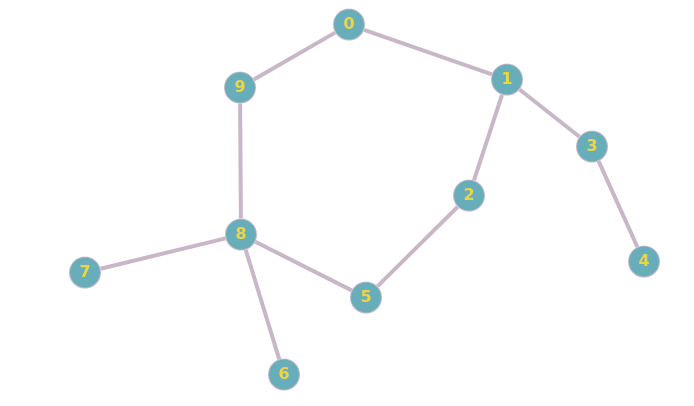
g.DFS(0, 7);

std::cout << '\n';

return 0;

}

**Результаты:**



Снимок экрана от 2021-04-26 20-24-05

**Вывод:**в ходе лабораторной работы были сформированы представления о методе и алгоритме поиска в глубину на графе и его применении при решении задач.