Министерство образования Республики Беларусь

Учреждения образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №1

По дисциплине МОИС за 1 семестр

«Нахождение компонент связности неориентированного графа»

Выполнил

Студент группы ИИ-21

Корнейчук А.И.

Проверил

Козинский А.А.

Брест 2022

**Задание 1-2:**Построить матрицу смежности и инцидентности для заданного графа. Изобразить граф.

Используя поиск в глубину и поиск в ширину написать программу, определяющую число компонент связности графа. Методы представляются в виде функций.

Вариант 6

(1,2),(1,3),(1,4), (2,3),(2,4),(3,4),(5,6),(6,7),(8,9)

Задание 1.

#include <iostream>

#include "../graph\_LIB.hh"

int main()

{

    convert c;

    std::string file\_path = "inf.txt";

    std::vector<int> nodes = c.reading\_file(file\_path);

    int max\_node = c.count\_of\_nodes(nodes);

    int n = nodes.size() / 2;

    std::cout<<n<<" "<<max\_node<<std::endl;

    std::vector<std::vector<int>> adjacencyMatrix = c.adjancy(nodes, max\_node);

    std::vector<std::vector<int>> incidenceMatrix = c.incidence(nodes, max\_node);

    for(int i=0;i<n\*2;i+=2){

        std::cout<<nodes[i]<<" ";

    }std::cout<<std::endl;

    for(int i=1;i<n\*2;i+=2){

        std::cout<<nodes[i]<<" ";

    }std::cout<<std::endl;

    std::cout<<"матрица смежности: "<<std::endl;

    for (int i = 0; i < max\_node; i++)

    {

        for (int j = 0; j < max\_node; j++)

        {

            std::cout << adjacencyMatrix[i][j]<<" ";

        }

        std::cout << std::endl;

    }

    std::cout << std::endl;

    std::cout<<"матрица инциндентности: "<<std::endl;

    for (int i = 0; i < max\_node; i++)

    {

        for (int j = 0; j < n; j++)

        {

            std::cout << incidenceMatrix[i][j]<<" ";

        }

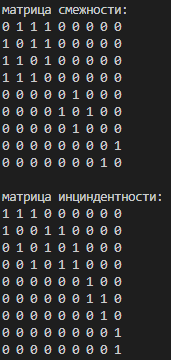
        std::cout << std::endl;

    }

return 0;

}

Вывод программы:



Задание 2.

#include <iostream>

#include "../graph\_LIB.hh"

int main(){

    convert c;

    std::string file\_path = "inf.txt";

    std::vector<int> nodes = c.reading\_file(file\_path);

    int max\_node = c.count\_of\_nodes(nodes);

    std::vector<std::vector<int>> adjacencyMatrix = c.adjancy(nodes, max\_node);

    int n = nodes.size()/2;

    alg search;

    std::vector<int> visited(max\_node);

    int count\_BFS;

    count\_BFS = search.find\_component\_with\_bfc(adjacencyMatrix);

    std::cout << "BFS: count\_BFS = " << count\_BFS << std::endl;

    //count\_BFS = search.conCompDFS(adjacencyMatrix,max\_node,n);

    /\*std::vector<int>\*/int count\_DFS;

    count\_DFS = search.conCompDFS(adjacencyMatrix);

    std::cout << "DFS: count\_DFS = " << count\_DFS << std::endl;

    /\*std::cout<<"DFS: count\_DFS = ";

    for(int i=0;i<count\_DFS.size();i++){

        std::cout<<count\_DFS[i]<<" ";

    }\*/

    return 0;

}

Вывод программы:



Граф:

