**Ministerul Educaţiei Tineretului şi Sportului al Republicii Moldova**

**Universitatea Tehnică a Moldovei**

**RAPORT**

**LA MATEMATCA DISCRETA**

Tema: ALGORITMUL DE CĂUTARE ÎN ADÂNCIME

Lucrarea de Laborator nr. 2

A efectuat Studentul grupei \_SI-212\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_Vozian Vladimir\_\_\_\_

*semnătura nume, prenume*

A verificat \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*nume, prenume profesor*

**Chisinău 2021**

1. **SCOPUL LUCRĂRII:**

* Studierea algoritmilor de căutare în graf şi a diferitor forme de păstrare şi prelucrare a datelor.
* Elaborarea procedurii de căutare în adâncime.

1. **SARCINA DE BAZĂ**
2. Elaboraţi procedura căutării în adâncime într-un graf arbitrar;
3. Elaboraţi un program cu următoarele posibilităţi:

* introducerea grafului în calculator,
* parcurgerea grafului în adâncime,
* vizualizarea rezultatelor la display şi imprimantă.

import numpy as np

import os

vizitate = []

def Adancimea(vizitate,lista,nodul):

    nodul-=1

    if nodul not in vizitate:

        print(nodul+1, end = " ")

        vizitate.append(nodul)

        for recurs in lista[nodul][0: len(lista[nodul])-1]:

            Adancimea(vizitate,lista,recurs)

#citire lista

def citireLista():

    nods = int(input("Cate noduri are matricea? "))

    lista = np.zeros(nods, dtype = object)

    for i in range(0, nods):

        l = []

        n = int(input("Cate legaturi sunt pentru nodul " + str(i+1) + ":"))

        if (n != 0):

            l = list(map(int,input("Dati legaturile : ").strip().split()))[:n]

            while not all(i <= nods for i in l):

                print("Date introduse gresit.")

                l = []

                l = list(map(int,input("Dati legaturile : ").strip().split()))[:n]

        l.append(0)

        lista[i] = l

    return lista

#afisare lista

def afisareLista(lista):

    nods = lista.shape[0]

    print("Lista adiacenta:")

    for i in range(1,nods+1):

        print(i,'- ',end='')

        for j in lista[i-1]:

            if(j != 0):

                print(str(j)+'\_',end = '')

            else:

                print(j)

lista = citireLista()

afisareLista(lista)

Adancimea(vizitate,lista,1)

## Exemple de executie a programului:

## 

## Matricele sunt introduse în forma de listă de incidență, rezultatul parcurgerii în adâncime fiind 1-2-5-4-3.

## 

## Deci având și reprezentarea prin desen a grafului, vedem ca într-adevăr este valid răspunsul programului pentru parcurgerea în adâncime.

## Și încă un exemplu mai complex:

## 

## 

## Concluzii:

Lucrarea dată era efectuată in python, sa exersat asupra cunoștiințelor parcurgerii grafurilor și anume parcurgerea grafului în adâncime cu efectuarea lui prin program, sa lucrat anume cu introducerea grafului prin lista adiacentă prin comoditatea ei.