Національний технічний університет України «КПІ ім. Ігоря Сікорського»

Факультет Інформатики та Обчислювальної Техніки

Кафедра Автоматизованих Систем Обробки Інформації та Управління

Лабораторна робота №1

з дисципліни «Дискретна математика»

на тему

«**Представлення графів**»

Виконав:

студент гр. ІС-91

Косюк Михайло

Викладач:

доц. Рибачук Л.В.

Київ – 2019

### Зміст

[Зміст 2](#_Toc190845267)

[1 Постановка задачі 3](#_Toc190845268)

[2 Результати виконання програми 4](#_Toc190845269)

[3 Лістинг програми](#_Toc190845271) 5

### 1 Постановка задачі

Ціль

Розглянути та вивчити різні типи представлення графів в пам’яті обчислювальних пристроїв (комп’ютерів).

Завдання

Реалізувати програмне застосування (програму), яке виконує наступні функції.

1. Зчитування графу з вхідного файлу. На вхід подається текстовий файл наступного вигляду:

n m

v1 u1

v2 u2

....

vm um

Тут *n* – кількість вершин графу (ціле число, більше нуля), *m* – кількість ребер графу (ціле число, більше нуля), *vi* та *ui* – початкова та кінцева вершина ребра *i* (1*vi**n*, 1*ui**n,* цілі числа). Індексація вершин у файлі ведеться з 1. Вважається, що граф є орієнтованим.

Таким чином можна сказати, що граф задається у файлі списком ребер.

2. Вивід матриць інцидентності та суміжності. За вимогою користувача програма повинна виводити матриці інцидентності та суміжності (окремі функції) на екран та/або у текстовий файл, який вказує користувач.

3. Визначити степінь вершин графу. За запитом користувача програма на екран та/або у файл виводить степінь усіх вершин графу. Визначити, чи граф є однорідним та якщо так, то вказати степінь однорідності графу.

4. Визначити всі висячі та ізольовані вершини. За запитом користувача програма на екран виводить перелік усіх висячих та ізольованих вершин графу.

Введення даних здійснюється у файл condition.txt, що знаходиться в одній директорії із фалом програми, згідно із заданим форматом.

Тестовий граф, що використовувався для обробки заданий такою послідовністю(граф, що містить 5 вершин і 7 ребер):

**5 7**

**1 2**

**2 3**

**2 2**

**5 1**

**4 1**

**4 5**

**5 5**

### 2 Результати виконання програми

### 

### 

### 3 Лістинг програми

from itertools import islice  
  
  
def print\_matrix(mat):  
 for r in mat:  
 for el in r:  
 print("%2d" % el, end=' ')  
 print()  
  
  
def gen\_adj\_mat(tops, e\_list):  
 adj\_mat = [0] \* tops  
 for i in range(tops):  
 adj\_mat[i] = [0] \* tops  
 for e in e\_list:  
 adj\_mat[e[0]-1][e[1]-1] = 1  
 print("\nМатриця суміжності: \n")  
 print\_matrix(adj\_mat)  
 return adj\_mat  
  
  
def gen\_inc\_mat(tops, edges, e\_list):  
 inc\_mat = [0] \* tops  
 for i in range(tops):  
 inc\_mat[i] = [0] \* edges  
 for t in range(len(inc\_mat)):  
 for e in range(len(inc\_mat[t])):  
 if e\_list[e][0] == e\_list[e][1] == t + 1:  
 inc\_mat[t][e] = 2  
 elif e\_list[e][0] == t + 1:  
 inc\_mat[t][e] = -1  
 elif e\_list[e][1] == t + 1:  
 inc\_mat[t][e] = 1  
 print("\nМатриця інцедентності: \n")  
 print\_matrix(inc\_mat)  
 return inc\_mat  
  
  
def get\_tops\_deg(tops, mat, prt):  
 mono = 0  
 l\_deg = 0  
 degrees = []  
 for t in range(tops):  
 deg = 0  
 for e in mat:  
 if t + 1 in e: deg += 1  
 if prt:  
 print("\nСтепінь вершини " + str(t + 1) + ": " + str(deg), end='')  
 mono += deg  
 if deg != 0: l\_deg = deg  
 degrees.append(deg)  
 if prt:  
 if degrees.count(l\_deg) == tops: print("\nГраф є однорідним, його степінь " + str(l\_deg))  
 else: print("\n Граф не є однорідним")  
 return degrees  
  
  
def check\_tops(degs):  
 print("\nКінцеві вершини: ", end='')  
 for t in range(len(degs)):  
 if degs[t] == 1: print(str(t + 1), end=', ')  
 print("\nІзольовані вершини: ", end='')  
 for t in range(len(degs)):  
 if degs[t] == 0: print(str(t + 1), end=', ')  
 print()  
  
  
file = open('condition.txt')  
tops\_num = int(file.read(1))  
edges\_num = int(file.read(3))  
edge\_list = []  
for edge in islice(file, 0, edges\_num):  
 ep = []  
 for top in edge.split(' '):  
 ep.append(int(top))  
 edge\_list.append(ep)  
print("Кількість вершин: " + str(tops\_num))  
print("Кількість ребер: " + str(edges\_num))  
print("Список граней: ", edge\_list)  
print("\nНатисніть 1 щоб вивести матрицю суміжності"  
 "\nНатисніть 2 щоб вивести матрицю інцедентності"  
 "\nНатисніть 3 щоб вивести степені вершин"  
 "\nНатисніть 4 щоб вивести ізольовані та кінцеві вершини"  
 "\nPress f щоб завешити роботу")  
control = ''  
degrees = []  
while control != 'f':  
 control = input("\n")  
 if control == '1': gen\_adj\_mat(tops\_num, edge\_list)  
 elif control == '2': gen\_inc\_mat(tops\_num, edges\_num, edge\_list)  
 elif control == '3': degrees = get\_tops\_deg(tops\_num, edge\_list, True)  
 elif control == '4':  
 degrees = get\_tops\_deg(tops\_num, edge\_list, False)  
 check\_tops(degrees)