Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №5

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Определение характеристик графов»

Выполнили:

студенты группы 22BВВ1

Митрошин Ю.Е

Коннов А.Д

Приняли:

Акифьев И.В

Юрова О.В

Пенза 2023

**Название**

Определение характеристик графов.

**Общие сведения.**

Если *G –* граф (рисунок 1), содержащий непустое множество *n* вершин *V* и множество ребер *E,* где *e(vi, vj) –* ребро между двумя произвольными вершинами *vi* и*vj*, тогда **размер** графа G есть мощность множества ребер |E(G)| или, количество ребер графа.

**Степенью**вершины графа *G* называется число инцидентных ей ребер. Степень вершины *vi* обозначается через *deg(vi).*

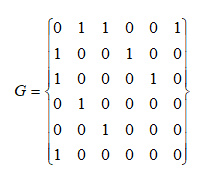
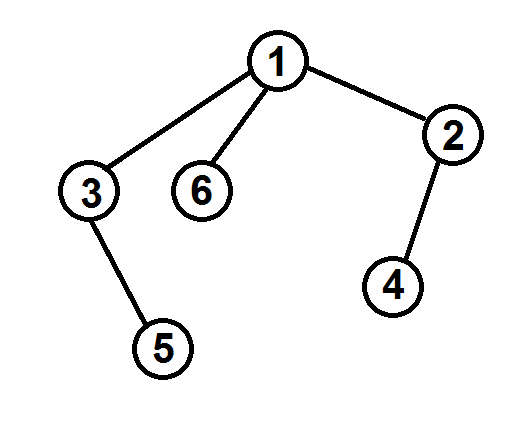


Рисунок 1 – Граф

Вершина *vi* со степенью 0 называется **изолированной***,* со степенью 1 – **концевой***.*

Вершина графа, смежная с каждой другой его вершиной, называется **доминирующей***.*

**Задание 1**

1. Сгенерируйте (используя генератор случайных чисел) матрицу смежности для неориентированного графа *G*. Выведите матрицу на экран.
2. Определите размер графа *G*, используя матрицу смежности графа.
3. Найдите изолированные, концевые и доминирующие вершины.

**Листинг**

import random

size = int(input("Введите размер: "))

matrix = []

for i in range(size):

row = []

for j in range(size):

row.append(0)

matrix.append(row)

for i in range(size):

for j in range(i + 1, size):

matrix[i][j] = matrix[j][i] = random.randint(0, 1)

for row in matrix:

print(row)

izol = []

end = []

dom = []

for i in range(size):

a = sum(matrix[i])

if a == 0:

izol.append(i)

elif a == 1:

end.append(i)

elif a == size - 1:

dom.append(i)

if not izol:

izol.append("Неть")

if not end:

end.append("Неть")

if not dom:

dom.append("Неть")

graphs = 0

for i in range(size):

for j in range(size):

graphs += matrix[i][j]

graphs = graphs // 2

inc = [[0] \* graphs for \_ in range (size)]

idx = 0

for j in range(size):

for i in range(j + 1, size):

if matrix[i][j] == 1:

inc[i][idx] = 1

inc[j][idx] = 1

idx += 1

izol1 = []

end1 = []

dom1 = []

for i in range(size):

c = sum(inc[i])

if c == 0:

izol1.append(i)

elif c == 1:

end1.append(i)

elif c == size - 1:

dom1.append(i)

if not izol1:

izol1.append("Неть")

if not end1:

end1.append("Неть")

if not dom1:

dom1.append("Неть")

print("Изолированные вершины: ", izol)

print("Концевые вершины: ", end)

print("Доминирующие вершины: ", dom)

print("Размер графа: ", graphs)

print("Матрица инцидентности:")

for b in range(size):

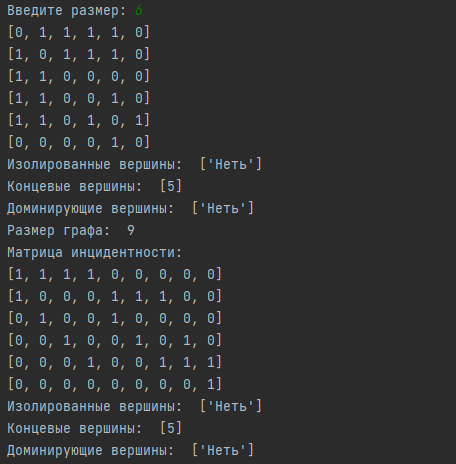
print(inc[b])

print("Изолированные вершины: ", izol1)

print("Концевые вершины: ", end1)

print("Доминирующие вершины: ", dom1)

**Результат работы программы:**

****

**Вывод:** в ходе лабораторной работы мы научились генерировать матрицу смежности для неориентированного графа G.