Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)
Кафедра компьютерных систем в управлении и проектировании (КСУП)

«ГЕНЕРАЦИЯ ГРАФА»

Отчет по лабораторной работе №1

По дисциплине «Дискретная математика»

Студенты гр. 573-2
Г.А Катренко.
С.С Масликов.
Т.Р Рафиков.
«19» апреля 2024 г.

Профессор каф. КСУП,
доктор техн. наук, доцент каф. КСУП
Д.В. Кручинин
«19» апреля 2024 г.

Томск 2024

Введение

Целью данной лабораторной работы является разработка программы для генерации матрицы смежности произвольной задаваемой размерности n и возможностью создания различных типов графов, таких как полные графы, графы с кратными ребрами и петлями. Для визуализации сгенерированных графов будут использоваться библиотеки NetworkX и MatPlotLib. Также в рамках лабораторной работы будет разработана функция преобразования матрицы смежности в матрицу инцидентности.

1 ОСНОВНАЯ ЧАСТЬ

Листинг:

```
import random

import networkx as nx

import matplotlib.pyplot as plt


#Функция генерации матрицы смежности

def generate_matrix(n, graf_type):

  matrix = [[0]*n for _ in range(n)]

  if graf_type == 'полный':

    for i in range(n):

      for j in range(n):

        if i !=j:

          matrix[i][j] = 1

  elif graf_type == 'кратные ребра и петли':

    for i in range(n):

      for j in range(n):

          matrix[i][j] = random.randint(0, 3)

  else:

    for i in range(n):

      for j in range(i, n):

        if i != j:

          matrix[i][j]=random.randint(0, 3)

        else:

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

  for i in range(n):

    for j in range(i):

      matrix[i][j] = matrix[j][i]

  return matrix
```

```
#Функция отрисовки графа
def draw_graf(adj_matrix):
  Graf = nx.Graph()
  n = len(adj_matrix)
  for i in range(n):
    Graf.add_node(i)
  for i in range(n):
    for j in range(n):
      if adj_matrix[i][j] > 0:
        Graf.add_edge(i, j, weight=adj_matrix[i][j])
  pos = nx.spring_layout(Graf)
  labels = nx.get_edge_attributes(Graf, 'weight')
  nx.draw(Graf, pos, with_labels=True, node_size=700, node_color='red', font_size=10, font_weight='bold')
  nx.draw_networkx_edge_labels(Graf, pos, edge_labels=labels)
  plt.show()
#Функция расчет матрицы инцедентности
def adjacency_to_incidence(adj_matrix):
  num_edges = 0
  num_vertex = len(adj_matrix)
  for i in range(num_vertex):
    for j in range(i+1, num_vertex):
      num_edges += adj_matrix[i][j]
  for i in range(num_vertex):
    num_edges += adj_matrix[i][i]
  incidence_matrix = [[0] * num_edges for _ in range(num_vertex)]
  edge_index=0
  for i in range(num_vertex):
    for j in range(i, num_vertex):
      if adj_matrix[i][j] != 0:
        for _ in range(adj_matrix[i][j]):
```

```
            incidence_matrix[i][edge_index] += 1
            incidence_matrix[j][edge_index] += 1
            edge_index += 1
    return incidence_matrix
#Вводим размерность и тип графа
n=int(input("Введите размерность матрицы: "))
graf_type = input("Выберите тип графа (случайный/полный/кратные ребра и петли): ").lower()
#Вывод матрицы смежности
adj_matrix = generate_matrix(n, graf_type)
print("Матрица смежности:")
name_column_adjacent = ' '
for i in range(n):
    name_column_adjacent += (" "+str(i)+" ")
print(name_column_adjacent)
name_row_adjacent = 0
for row in adj_matrix:
    print(str(name_row_adjacent) + str(row))
    name_row_adjacent +=1
#Вывод матрицы инцедентности
incidence_matrix = adjacency_to_incidence(adj_matrix)
print("Матрица инцидентности:")
name_row_incidence=0
for row in incidence_matrix:
    print(str(name_row_incidence)+str(row))
    name_row_incidence+=1
draw_graf(adj_matrix)
```

Заключение

В ходе лабораторной работы мы разработали программу в языке программирования Python, способную генерировать матрицы смежности для графов различных типов и размерностей. Благодаря использованию библиотеки NetworkX удалось эффективно визуализировать сгенерированные графы, что облегчило их анализ. Кроме того, функция преобразования матрицы смежности в матрицу инцидентности позволяет удобно работать с графами в различных представлениях.