

Министерство высшего образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего
образования
**«Пермский национальный исследовательский политехнический
университет» (ПНИПУ)**
Электротехнический факультет
Кафедра «Информационные технологии и автоматизированные системы»

ОТЧЁТ
Дискретная математика
«Определение компонент связности графа»

Выполнил
Студент группы РИС-22-26
Прядеин И.А.
Проверил доцент кафедры
ИТАС
Рустамханова Г. И.

Пермь 2024

Постановка задачи:

Дана матрица смежности неориентированного графа, состоящего из 10 вершин.

Определить компоненты связности графа.

Алгоритм работы:

Считывание файла (рис. 1) реализовано с помощью функции “fromfile” из библиотеки “numpy”, результат считывания которой присваивается переменной “matrix”, хранящей исходную матрицу.

```
matrix = np.ndarray(shape=(10, 10), buffer=np.fromfile(f'g1{fileNumber}.txt', sep=' '))
boolMatrix = np.array(matrix, dtype=bool)
print('Исходная матрица:')
print(pd.DataFrame(np.where(boolMatrix, 1, 0)))
```

Рис.1 – Считывание матрицы из файла

Следующим шагом рассчитывается матрица достижимости (рис.2) используя формулу, показанную на (рис. 3).

```
resultMatrix = np.eye(len(matrix), dtype=bool)
tempMatrix = boolMatrix
for i in range(0, 9):
    tempMatrix = np.matmul(resultMatrix, boolMatrix)
    resultMatrix += tempMatrix
```

Рис. 2 – Вычисление матрицы достижимости

$$\mathbf{R(G)} = \mathbf{E} \vee \mathbf{A} \vee \mathbf{A}^2 \vee \dots \vee \mathbf{A}^{n-1}$$

Рис. 3 – Формула вычисления матрицы достижимости

Далее определяются компоненты связности графа (рис. 4).

```
df = pd.DataFrame(resultMatrix)
result = []
while not df.empty:
    indices = np.where(df.iloc[0])[0]
    result.append(set(indices))
    print(df)
    df = df.drop(indices)
```

Рис. 4 – Определение компонент связности

Результат тестирования:

Исходная матрица:

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	1	0	0	0	1	0
1	0	0	0	0	0	1	0	0	0	1
2	0	0	0	0	1	0	0	0	0	0
3	1	0	0	0	0	0	1	0	0	0
4	0	0	1	0	0	0	0	1	0	0
5	0	1	0	0	0	0	0	0	0	1
6	0	0	0	1	0	0	0	0	1	0
7	0	0	0	0	1	0	0	0	0	0
8	1	0	0	0	0	0	1	0	0	0
9	0	1	0	0	0	1	0	0	0	0

```
resultMatrix = np.eye(len(matrix), dtype=bool)
tempMatrix = boolMatrix
for i in range(0, 9):
    tempMatrix = np.matmul(resultMatrix, boolMatrix)
    resultMatrix += tempMatrix
# print('Матрица достижимости:')
# print(pd.DataFrame(np.where(resultMatrix, 1, 0)))
```

```
df = pd.DataFrame(resultMatrix)
result = []
while not df.empty:
    indices = np.where(df.iloc[0])[0]
    result.append(set(indices))
    df = df.drop(indices)

print('Компоненты:')
for i in range(len(result)):
    print(f'{i + 1}: {result[i]}')
```

Компоненты:

1: {0, 8, 3, 6}
2: {1, 5, 9}
3: {2, 4, 7}

Рис. 5 – Файл “g11”

Исходная матрица:

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	1	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0
3	0	0	0	0	1	0	0	0	0	0
4	0	0	0	1	0	0	0	1	0	0
5	0	1	1	0	0	0	0	0	1	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	1	0	0	0	0	0
8	1	0	0	0	0	1	0	0	0	1
9	0	0	0	0	0	0	0	0	1	0

```
resultMatrix = np.eye(len(matrix), dtype=bool)
tempMatrix = boolMatrix
for i in range(0, 9):
    tempMatrix = np.matmul(resultMatrix, boolMatrix)
    resultMatrix += tempMatrix
# print('Матрица достижимости:')
# print(pd.DataFrame(np.where(resultMatrix, 1, 0)))
```

```
df = pd.DataFrame(resultMatrix)
result = []
while not df.empty:
    indices = np.where(df.iloc[0])[0]
    result.append(set(indices))
    df = df.drop(indices)

print('Компоненты:')
for i in range(len(result)):
    print(f'{i + 1}: {result[i]}')
```

Компоненты:

1: {0, 1, 2, 5, 8, 9}
2: {3, 4, 7}
3: {6}

Рис. 6 – Файл “g12”

Исходная матрица:

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	1	1	1
1	0	0	0	0	0	0	1	0	0	0
2	0	0	0	1	0	0	0	0	0	1
3	0	0	1	0	1	0	0	0	0	0
4	0	0	0	1	0	1	0	0	0	0
5	0	1	0	0	1	0	0	0	0	0
6	1	0	0	0	0	0	0	1	1	0
7	1	0	0	0	0	0	1	0	1	0
8	1	0	0	0	0	0	1	1	0	0
9	0	1	1	0	0	0	0	0	0	0

```
resultMatrix = np.eye(len(matrix), dtype=bool)
tempMatrix = boolMatrix
for i in range(0, 9):
    tempMatrix = np.matmul(resultMatrix, boolMatrix)
    resultMatrix += tempMatrix
# print('Матрица достижимости:')
# print(pd.DataFrame(np.where(resultMatrix, 1, 0)))
```

```
df = pd.DataFrame(resultMatrix)
result = []
while not df.empty:
    indices = np.where(df.iloc[0])[0]
    result.append(set(indices))
    df = df.drop(indices)

print('Компоненты:')
for i in range(len(result)):
    print(f'{i + 1}: {result[i]}')
```

Компоненты:

1: {0, 8, 6, 7}

2: {1, 2, 3, 4, 5, 9}

Рис. 7 – Файл “g13”

Исходный код:

```
import numpy as np
import pandas as pd

def IntInput(message):
    number = None
    while number is None:
        try:
            number = int(input(message))
        except ValueError:
            print('Некорректное число!')
    return number

fileNumber = IntInput('Номер файла: ')
matrix = np.ndarray(shape=(10, 10), buffer=np.fromfile(f'gl {fileNumber}.txt',
sep=' '))
boolMatrix = np.array(matrix, dtype=bool)
print('Исходная матрица:')
print(pd.DataFrame(np.where(boolMatrix, 1, 0)))
resultMatrix = np.eye(len(matrix), dtype=bool)
tempMatrix = boolMatrix
for i in range(0, 9):
    tempMatrix = np.matmul(resultMatrix, boolMatrix)
    resultMatrix += tempMatrix
# print('Матрица достижимости:')
# print(pd.DataFrame(np.where(resultMatrix, 1, 0)))
df = pd.DataFrame(resultMatrix)
```

```
result = []

while not df.empty:
    indices = np.where(df.iloc[0])[0]
    result.append(set(indices))
    df = df.drop(indices)

print('Компоненты:')
for i in range(len(result)):
    print(f'{i + 1}: {result[i]}')
```