

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

Дисциплина: «Математические методы теории систем»
Лабораторная работа № 2
на тему: ««Описание потоков информации в системах»»

Выполнил: студент группы АСУ4-22-1м
Попов Кирилл Михайлович
Проверил: ассистент кафедры ИТАС
Тютюных Артём Александрович

ЦЕЛЬ РАБОТЫ

Анализ графов потока информации, определение их свойств, построение алгоритма определения свойств.

ЗАДАНИЕ

Задача 1. Дано описание системы на рис. 1.

1. Формально выявить все свойства данного информационного графа.
2. Реализовать решение задачи на одном из языков программирования.
3. Привести результаты работы программы.

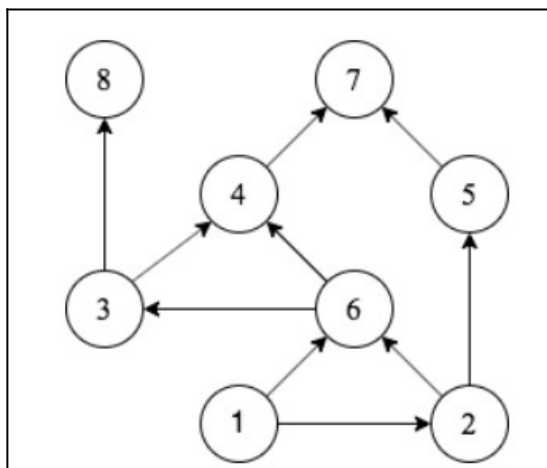


Рис. 1. Описание системы

Задача 2. Выбранный вариант системы приведён на рис. 2.

Выявить все свойства одного информационного графа из предложенных ниже вариантов, используя разработанную программу. Привести результаты работы программы

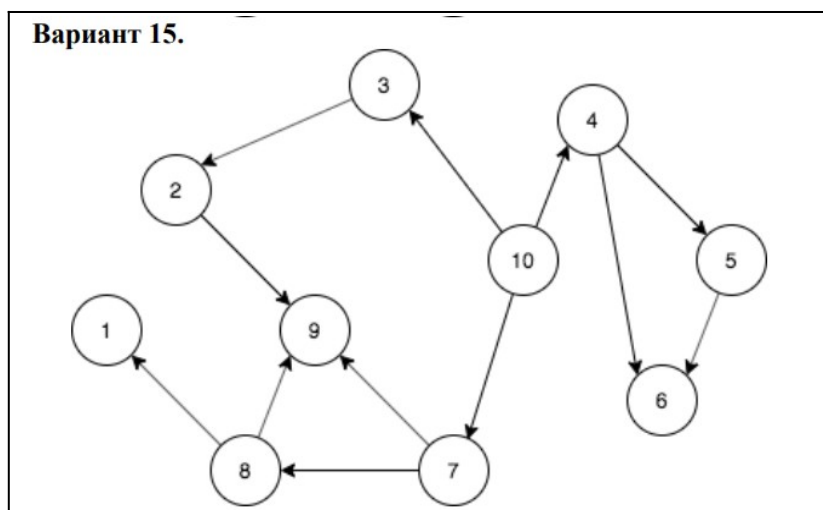


Рис. 2. Описание системы. Вариант 15

ВЫПОЛНЕНИЕ РАБОТЫ

Задание 1

Получение матрицы:

Таблица 1. Матрица смежности

i\j	1	2	3	4	5	6	7	8
1	0	1	0	0	0	1	0	0
2	0	0	0	0	1	1	0	0
3	0	0	0	1	0	0	0	1
4	0	0	0	0	0	0	1	0
5	0	0	1	1	0	0	1	0
6	0	0	1	1	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

Определение длины путей графа:

Далее матрица смежности возводится в степень для вычисления длины путей графа. q_j и q_i есть сумма столбцов и строк соответственно. Для удобства анализа данные выделены различными цветами:

- жёлтый — от узла до узла 1 путь длиной λ ;
- синий — от узла до узла имеются 2 различных пути длиной λ ;
- и т. д.

Таблица 2. Элементы 0 порядка (A^0)

i\j	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
q_i	1	1	1	1	1	1	1	1

Таблица 3. Элементы 1 порядка (A^1)

i\j	1	2	3	4	5	6	7	8	q_i
1	0	1	0	0	0	1	0	0	2
2	0	0	0	0	1	1	0	0	2
3	0	0	0	1	0	0	0	1	2
4	0	0	0	0	0	0	1	0	1
5	0	0	0	0	0	0	1	0	1
6	0	0	1	1	0	0	0	0	2
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
q_j	0	1	1	2	1	2	2	1	

Таблица 4. Элементы 2 порядка (A^2)

i\j	1	2	3	4	5	6	7	8	q_i
1	0	0	1	1	1	1	0	0	4
2	0	0	1	1	0	0	1	0	3
3	0	0	0	0	0	0	1	0	1
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	1	0	0	1	1	3
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
q_j	0	0	2	3	1	1	3	1	

Таблица 5. Элементы 3 порядка (A^3)

i\j	1	2	3	4	5	6	7	8	q_i
1	0	0	1	2	0	0	2	1	0
2	0	0	0	1	0	0	1	1	0
3	0	0	0	0	0	0	0	0	1
4	0	0	0	0	0	0	0	0	3
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	1	0	0
7	0	0	0	0	0	0	0	0	4
8	0	0	0	0	0	0	0	0	2
q_j	0	0	1	3	0	0	4	2	

Таблица 6. Элементы 4 порядка (A^4)

i\j	1	2	3	4	5	6	7	8	q_i
1	0	0	0	1	0	0	2	1	4
2	0	0	0	0	0	0	1	0	1
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
q_j	0	0	0	1	0	0	3	1	

Таблица 7. Элементы 5 порядка (A^5)

i\j	1	2	3	4	5	6	7	8	q_i
1	0	0	0	0	0	0	1	0	1
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
q_j	0	0	0	0	0	0	1	0	

Следующие матрицы являются нулевыми, следовательно достигнута максимальная длина графа, которая равна 5.

Далее приведена сумма всех матриц.

Таблица 8. Матрица достижимости

i\j	1	2	3	4	5	6	7	8	q_i
1	0	1	2	4	1	2	5	2	17
2	0	0	1	2	1	1	3	1	9
3	0	0	0	1	0	0	1	1	3
4	0	0	0	0	0	0	1	0	1
5	0	0	0	0	0	0	1	0	1
6	0	0	1	2	0	0	2	1	6
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
q_j	0	1	4	9	2	3	13	5	

Исходя из матрицы достижимости максимальная длина пути графа от узла 1 до 7 равна 5.

Результаты работы программы приведены в приложении А.

Задание 2

Результаты работы программы приведены в приложении Б.

ПРИЛОЖЕНИЕ А

Получение матриц

```
... m_smeg^0:
      [1, 1, 1, 1, 1, 1, 1, 1]

m_smeg^1:
      [[0 1 0 0 0 1 0 0]
      [0 0 0 0 1 1 0 0]
      [0 0 0 1 0 0 0 1]
      [0 0 0 0 0 0 1 0]
      [0 0 0 0 0 0 1 0]
      [0 0 1 1 0 0 0 0]
      [0 0 0 0 0 0 0 0]
      [0 0 0 0 0 0 0 0]]
Q_j = [0, 1, 1, 2, 1, 2, 2, 1]
Q_i = [2, 2, 2, 1, 1, 2, 0, 0]

m_smeg^2:
      [[0 0 1 1 1 1 0 0]
      [0 0 1 1 0 0 1 0]
      [0 0 0 0 0 0 1 0]
      [0 0 0 0 0 0 0 0]
      [0 0 0 0 0 0 0 0]
      [0 0 0 1 0 0 1 1]
      [0 0 0 0 0 0 0 0]
      [0 0 0 0 0 0 0 0]]
Q_j = [0, 0, 2, 3, 1, 1, 3, 1]
Q_i = [4, 3, 1, 0, 0, 3, 0, 0]

m_smeg^3:
      [[0 0 1 2 0 0 2 1]
      [0 0 0 1 0 0 1 1]
      [0 0 0 0 0 0 0 0]
      [0 0 0 0 0 0 0 0]
      [0 0 0 0 0 0 0 0]
      [0 0 0 0 0 0 1 0]
      [0 0 0 0 0 0 0 0]
      [0 0 0 0 0 0 0 0]]
Q_j = [0, 0, 1, 3, 0, 0, 4, 2]
Q_i = [6, 3, 0, 0, 0, 1, 0, 0]
```

```

m_smeg^5:
[[0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]]
Q_j = [0, 0, 0, 0, 0, 0, 1, 0]
Q_i = [1, 0, 0, 0, 0, 0, 0, 0]

```

```

m_smeg^6:
[[0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]]
Q_j = [0, 0, 0, 0, 0, 0, 0, 0]
Q_i = [0, 0, 0, 0, 0, 0, 0, 0]

```

```

total:
[[0 1 2 4 1 2 5 2]
 [0 0 1 2 1 1 3 1]
 [0 0 0 1 0 0 1 1]
 [0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 1 0]
 [0 0 1 2 0 0 2 1]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]]
Q_j = [0, 1, 4, 9, 2, 3, 13, 5]
Q_i = [17, 9, 3, 1, 1, 6, 0, 0]

```

Элементы 0 порядка

[1, 2, 3, 4, 5, 6, 7, 8]

[1, 0, 0, 0, 0, 0, 0, 0]

Элементы 1 порядка

[0, 2, 3, 4, 5, 6, 7, 8]

[1, 2, 0, 0, 0, 0, 0, 0]

Элементы 2 порядка

[0, 0, 3, 4, 5, 6, 7, 8]

[1, 2, 0, 0, 5, 6, 0, 0]

Элементы 3 порядка

[0, 0, 3, 4, 0, 0, 7, 8]

[1, 2, 3, 0, 5, 6, 0, 0]

Элементы 4 порядка

[0, 0, 0, 4, 0, 0, 7, 8]

[1, 2, 3, 4, 5, 6, 0, 8]

Элементы 5 порядка

[0, 0, 0, 0, 0, 0, 7, 0]

[1, 2, 3, 4, 5, 6, 7, 8]

узел: [1, 2, 3, 4, 5, 6, 7, 8]

порядок: [0, 1, 3, 4, 2, 2, 5, 4]

2. Определение тактности системы

```
print('Тактность == ', max(res))
```

[162]

```
... Тактность = 5
```

3. Определение контуров

```
def search_contour():  
    s = -1  
    flag_1 = False  
    for a in range(0,8):  
        s = s + 1  
        if total[s][a] > 0:  
            flag_1 = True  
  
    if flag_1 == False:  
        print('Контур не найден')  
    else:  
        print('Контур найден')  
  
search_contour()
```

[163]

```
... Контур не найден
```

4. Определение входных элементов потока

```
[164] for i in range(8):
      | if Q_j_1[i] == 0:
      |     print('Входной элемент:', i+1 )
Python
```

... Входной элемент: 1

5. Определение выходных элементов потока

```
[165] for i in range(8):
      | if Q_i_1[i] == 0:
      |     print('Выходной элемент:', i+1 )
Python
```

... Выходной элемент: 7
Выходной элемент: 8

6. Определение висящих вершин

```
def searsh_v():
    flag = False
    for i in range(8):
        if Q_i_1[i] == 0:
            if Q_j_1[i] == 0:
                flag = True
    if flag == True:
        print('Висящая вершина найдена')
    else:
        print('Висящих вершин нет')
searsh_v()
```

[166]

Python

*** Висящих вершин нет

7. Определение числа путей длиной Lambda

def count_path(matr, str): ...

 10 элементов длиной 1 с 1 путём
 11 элементов длиной 2 с 1 путём
 6 элементов длиной 3 с 1 путём
 2 элементов длиной 3 с 2 путями
 3 элементов длиной 4 с 1 путём
 1 элементов длиной 4 с 2 путями
 1 элементов длиной 5 с 1 путём

8. Определение всевозможных путей между двумя элементами.

```
for k in range(8):
    for i in range(8):
        if total[k][i] > 0:
            print('От узла',k+1,'до узла',i+1,'количество путей =', total[k][i])
```

[168]

Python

```
... От узла 1 до узла 2 количество путей = 1
    От узла 1 до узла 3 количество путей = 2
    От узла 1 до узла 4 количество путей = 4
    От узла 1 до узла 5 количество путей = 1
    От узла 1 до узла 6 количество путей = 2
    От узла 1 до узла 7 количество путей = 5
    От узла 1 до узла 8 количество путей = 2
    От узла 2 до узла 3 количество путей = 1
    От узла 2 до узла 4 количество путей = 2
    От узла 2 до узла 5 количество путей = 1
    От узла 2 до узла 6 количество путей = 1
    От узла 2 до узла 7 количество путей = 3
    От узла 2 до узла 8 количество путей = 1
    От узла 3 до узла 4 количество путей = 1
    От узла 3 до узла 7 количество путей = 1
    От узла 3 до узла 8 количество путей = 1
    От узла 4 до узла 7 количество путей = 1
    От узла 5 до узла 7 количество путей = 1
    От узла 6 до узла 3 количество путей = 1
    От узла 6 до узла 4 количество путей = 2
    От узла 6 до узла 7 количество путей = 2
    От узла 6 до узла 8 количество путей = 1
```

9. Определение всех элементов, участвующих в формировании данного

```

s1 = set()
s2 = set()
s3 = set()
s4 = set()
s5 = set()
s6 = set()
s7 = set()
s8 = set()
s = [s1, s2, s3, s4, s5, s6, s7, s8]

for k in range(8):
    for i in range(8):
        if total[i][k] > 0:
            s[k].add(i+1)
for i in range(8):
    print('Элементы формирующие узел', i+1, ': ', s[i])

```

[344] Python

```

... Элементы формирующие узел 1 : set()
Элементы формирующие узел 2 : {1}
Элементы формирующие узел 3 : {1, 2, 6}
Элементы формирующие узел 4 : {1, 2, 3, 6}
Элементы формирующие узел 5 : {1, 2}
Элементы формирующие узел 6 : {1, 2}
Элементы формирующие узел 7 : {1, 2, 3, 4, 5, 6}
Элементы формирующие узел 8 : {1, 2, 3, 6}

```

ПРИЛОЖЕНИЕ Б

Листинг 1. Вывод программы

```
PS C:\Users\Кирилл\Desktop\учеба\MMTS\lab_2> &  
C:/Users/Кирилл/AppData/Local/Microsoft/WindowsApps/python3.10.exe  
c:/Users/Кирилл/Desktop/учеба/MMTS/lab_2/work_2.py
```

ЗАДАНИЕ 2

```
m_smeg^0:  
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]  
  
m_smeg^1:  
[[0 0 0 0 0 0 0 0 0 0]  
[1 0 1 0 0 0 0 0 0 1]  
[0 0 0 1 0 0 0 0 0 0]  
[0 0 0 0 1 0 0 0 0 1]  
[0 0 0 0 0 1 1 0 0 0]  
[0 0 0 0 0 0 0 0 0 0]  
[0 0 0 0 0 1 0 0 0 0]  
[1 0 0 0 0 0 1 0 0 0]  
[0 1 0 0 0 0 0 1 0 0]  
[0 0 0 0 0 0 0 0 0 0]]  
Q_j = [2, 1, 1, 1, 1, 2, 2, 1, 0, 2]  
Q_i = [0, 3, 1, 2, 2, 0, 1, 2, 2, 0]  
  
m_smeg^2:  
[[0 0 0 0 0 0 0 0 0 0]  
[0 0 0 1 0 0 0 0 0 0]  
[0 0 0 0 1 0 0 0 0 1]  
[0 0 0 0 0 1 1 0 0 0]  
[0 0 0 0 0 1 0 0 0 0]  
[0 0 0 0 0 0 0 0 0 0]  
[0 0 0 0 0 0 0 0 0 0]  
[0 0 0 0 0 1 0 0 0 0]  
[2 0 1 0 0 0 1 0 0 1]  
[0 0 0 0 0 0 0 0 0 0]]  
Q_j = [2, 0, 1, 1, 1, 3, 2, 0, 0, 2]  
Q_i = [0, 1, 2, 2, 1, 0, 0, 1, 5, 0]  
  
m_smeg^3:  
[[0 0 0 0 0 0 0 0 0 0]  
[0 0 0 0 1 0 0 0 0 1]  
[0 0 0 0 0 1 1 0 0 0]  
[0 0 0 0 0 1 0 0 0 0]  
[0 0 0 0 0 0 0 0 0 0]  
[0 0 0 0 0 0 0 0 0 0]  
[0 0 0 0 0 0 0 0 0 0]  
[0 0 0 0 0 0 0 0 0 0]  
[0 0 0 0 0 0 0 0 0 0]  
[0 0 0 1 0 1 0 0 0 0]  
[0 0 0 0 0 0 0 0 0 0]]  
Q_j = [0, 0, 0, 1, 1, 3, 1, 0, 0, 1]  
Q_i = [0, 2, 2, 1, 0, 0, 0, 0, 2, 0]  
  
m_smeg^4:  
[[0 0 0 0 0 0 0 0 0 0]  
[0 0 0 0 0 1 1 0 0 0]  
[0 0 0 0 0 1 0 0 0 0]  
[0 0 0 0 0 0 0 0 0 0]  
[0 0 0 0 0 0 0 0 0 0]  
[0 0 0 0 0 0 0 0 0 0]  
[0 0 0 0 0 0 0 0 0 0]  
[0 0 0 0 0 0 0 0 0 0]]
```

Листинг 1. Вывод программы (продолжение)

```
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 1 0 0 0 0 1]
[0 0 0 0 0 0 0 0 0 0]]
Q_j = [0, 0, 0, 0, 1, 2, 1, 0, 0, 1]
Q_i = [0, 2, 1, 0, 0, 0, 0, 0, 2, 0]
```

```
m_smeg^5:
[[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 1 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 1 1 0 0 0]
[0 0 0 0 0 0 0 0 0 0]]
Q_j = [0, 0, 0, 0, 0, 2, 1, 0, 0, 0]
Q_i = [0, 1, 0, 0, 0, 0, 0, 0, 2, 0]
```

```
m_smeg^6:
[[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 1 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]]
Q_j = [0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
Q_i = [0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
```

```
m_smeg^7:
[[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0]]
Q_j = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Q_i = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

```
total:
[[0 0 0 0 0 0 0 0 0 0]
[1 0 1 1 1 2 1 0 0 2]
[0 0 0 1 1 2 1 0 0 1]
[0 0 0 0 1 2 1 0 0 1]
[0 0 0 0 0 2 1 0 0 0]
[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 1 0 0 0 0]
[1 0 0 0 0 1 1 0 0 0]
[2 1 1 1 1 3 2 1 0 2]
[0 0 0 0 0 0 0 0 0 0]]
Q_j = [4, 1, 2, 3, 4, 13, 7, 1, 0, 6]
```

Листинг 1. Вывод программы (продолжение)

```
Q_i = [0, 9, 6, 5, 3, 0, 1, 3, 14, 0]
```

СВОЙСТВА:

1. Определение порядка элементов:

Элементы 0 порядка

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
[0, 0, 0, 0, 0, 0, 0, 0, 9, 0]
```

Элементы 1 порядка

```
[1, 2, 3, 4, 5, 6, 7, 8, 0, 10]  
[0, 2, 0, 0, 0, 0, 0, 8, 9, 0]
```

Элементы 2 порядка

```
[1, 0, 3, 4, 5, 6, 7, 0, 0, 10]  
[1, 2, 3, 0, 0, 0, 0, 8, 9, 0]
```

Элементы 3 порядка

```
[0, 0, 0, 4, 5, 6, 7, 0, 0, 10]  
[1, 2, 3, 4, 0, 0, 0, 8, 9, 0]
```

Элементы 4 порядка

```
[0, 0, 0, 0, 5, 6, 7, 0, 0, 10]  
[1, 2, 3, 4, 5, 0, 0, 8, 9, 10]
```

Элементы 5 порядка

```
[0, 0, 0, 0, 0, 6, 7, 0, 0, 0]  
[1, 2, 3, 4, 5, 0, 7, 8, 9, 10]
```

Элементы 6 порядка

```
[0, 0, 0, 0, 0, 6, 0, 0, 0, 0]  
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

узел: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

порядок: [2, 1, 2, 3, 4, 6, 5, 1, 0, 4]

2. Опредедение тактности системы:

Тактность = 6

3. Определение контуров

Контуры не найдены

4. Определение входных элементов потока

Входной элемент: 9

5. Определение выходных элементов потока

Выходной элемент: 1

Выходной элемент: 6

Выходной элемент: 10

6. Определение висящих вершин

Висящих вершин нет

7. Определение числа путей длиной Lambda

13 элементов длиной 1 с 1 путём

10 элементов длиной 2 с 1 путём

От узла 9 до узла 10 количество путей = 2

8. Определение всевозможных путей между двумя элементами

От узла 2 до узла 9 количество путей = 1
От узла 3 до узла 2 количество путей = 1
От узла 3 до узла 9 количество путей = 1
От узла 4 до узла 5 количество путей = 1
От узла 4 до узла 6 количество путей = 2
От узла 5 до узла 6 количество путей = 1
От узла 7 до узла 1 количество путей = 1
От узла 7 до узла 8 количество путей = 1
От узла 7 до узла 9 количество путей = 2
От узла 8 до узла 1 количество путей = 1
От узла 8 до узла 9 количество путей = 1
От узла 10 до узла 1 количество путей = 1
От узла 10 до узла 2 количество путей = 1
От узла 10 до узла 3 количество путей = 1
От узла 10 до узла 4 количество путей = 1
От узла 10 до узла 5 количество путей = 1
От узла 10 до узла 6 количество путей = 2
От узла 10 до узла 7 количество путей = 1
От узла 10 до узла 8 количество путей = 1
От узла 10 до узла 9 количество путей = 3

9. Определение всех элементов, участвующих в формировании данного

Элементы формирующие узел 1 : {8, 10, 7}
Элементы формирующие узел 2 : {10, 3}
Элементы формирующие узел 3 : {10}
Элементы формирующие узел 4 : {10}
Элементы формирующие узел 5 : {10, 4}
Элементы формирующие узел 6 : {10, 4, 5}
Элементы формирующие узел 7 : {10}
Элементы формирующие узел 8 : {10, 7}
Элементы формирующие узел 9 : {2, 3, 7, 8, 10}
Элементы формирующие узел 10 : set()

ЗАКЛЮЧЕНИЕ

Приведены результаты работы программ, написанных на python и в среде Jupiter. Т. к. при копировании кода из Jupiter слетает форматирование, приведены скриншоты. Программный код: https://github.com/Kirpo97/MMTS_labs/tree/main/lab_2.