|  |  |
| --- | --- |
| КГЭУ | МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  **Федеральное государственное бюджетное образовательное учреждение высшего образования**  **«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»**  (ФГБОУ ВО «КГЭУ») |

# Институт цифровых технологий и экономики Кафедра «Цифровые системы и модели»



**Дисциплина «Теория систем и системный анализ» Отчет по практической работе № 3**

# Тема: «Полные циклы и полные графы. Алгоритм перечисления полных простых циклов полного графа. Задача коммивояжера»

**Исполнитель**: Соловьёв Л.А,

**Группа**: ПИ-1-22

# Вариант: № 17

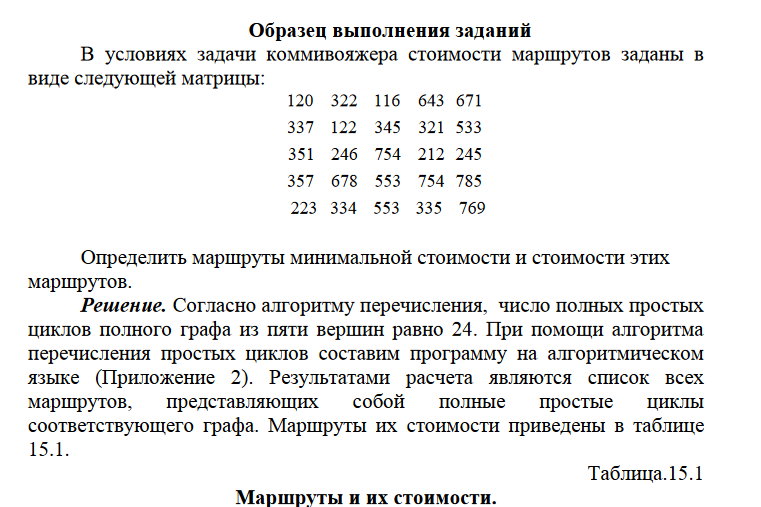
**Проверил:** доц. Абдулмянов Т.Р.

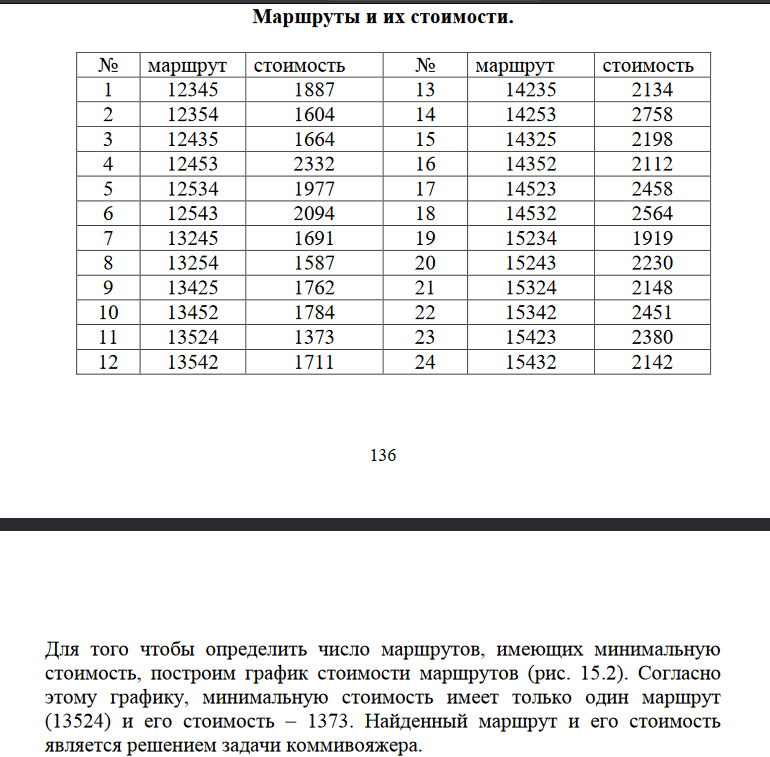
# КАЗАНЬ – 2024

**Цель работы:** Составить программу перечисления всех простых циклов полного графа. На алгоритмическом языке составить программу решения задачи коммивояжера и решить задачу для заданных стоимостей маршрутов.

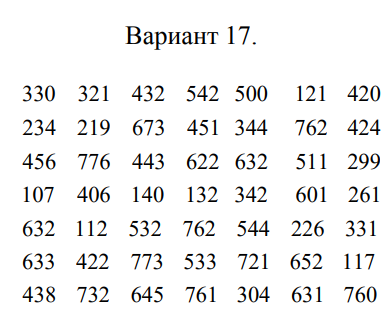
**Краткая теория**

Циклом данного графа называется последовательность ребер, которая начинается с некоторой вершины и завершается на этой же вершине. Цикл называется простым, если цикл не имеет повторяющихся ребер. Полным простым циклом графа G называются простые циклы графа G, содержащие все его вершины. Граф называется полным, если каждая его вершина связана ребром с каждой другой вершиной данного графа. Рассмотрим задачу перечисления всех полных простых циклов графа G. Решение задачи перечисления полных циклов дает алгоритм оптимизации маршрутов в некоторых сетевых задачах.







****

**Решение задачи коммивояжера**

Sub commiw()

Dim a(8, 8)

a(1, 1) = 330: a(1, 2) = 321: a(1, 3) = 432: a(1, 4) = 542: a(1, 5) = 500: a(1, 6) = 121: a(1, 7) = 420

a(2, 1) = 234: a(2, 2) = 219: a(2, 3) = 673: a(2, 4) = 451: a(2, 5) = 344: a(2, 6) = 762: a(2, 7) = 424

a(3, 1) = 456: a(3, 2) = 776: a(3, 3) = 443: a(3, 4) = 622: a(3, 5) = 632: a(3, 6) = 511: a(3, 7) = 299

a(4, 1) = 107: a(4, 2) = 406: a(4, 3) = 140: a(4, 4) = 132: a(4, 5) = 342: a(4, 6) = 601: a(4, 7) = 261

a(5, 1) = 632: a(5, 2) = 112: a(5, 3) = 532: a(5, 4) = 762: a(5, 5) = 544: a(5, 6) = 226: a(5, 7) = 331

a(6, 1) = 633: a(6, 2) = 422: a(6, 3) = 773: a(6, 4) = 533: a(6, 5) = 721: a(6, 6) = 652: a(6, 7) = 117

a(7, 1) = 438: a(7, 2) = 732: a(7, 3) = 645: a(7, 4) = 761: a(7, 5) = 304: a(7, 6) = 631: a(7, 7) = 760

m = 1

For i = 2 To 7

For j = 2 To 7

For k = 2 To 7

For l = 2 To 7

For m = 2 To 7

For n = 2 To 7

If i = j Then GoTo 10

If i = k Then GoTo 10

If i = l Then GoTo 10

If i = m Then GoTo 10

If i = n Then GoTo 10

If j = k Then GoTo 10

If j = l Then GoTo 10

If j = m Then GoTo 10

If j = n Then GoTo 10

If k = l Then GoTo 10

If k = m Then GoTo 10

If k = n Then GoTo 10

If l = m Then GoTo 10

If l = n Then GoTo 10

If m = n Then GoTo 10

s1 = a(1, i) + a(i, j) + a(j, k) + a(k, l) + a(l, m) + a(m, n) + a(n, 1)

s2 = 1000000 + 100000 \* i + 10000 \* j + 1000 \* k + 100 \* l + 10 \* m + n

Cells(m, 1) = s1

Cells(m, 2) = s2

m = m + 1

10

Next n, m, l, k, j, i

End Sub

Для того чтобы определить число маршрутов, имеющих минимальную стоимость, отсортируем диапазон данных сначала по y(стоимость маршрута), затем по x(маршрут). Минимальную стоимость имеет только один маршрут (1675243) и его стоимость – 1701. Найденный маршрут и его стоимость является решением задачи коммивояжер



**Решение задачи на языке Python:**

Для решения задачи необходимо воспользоваться библиотекой итертулс для перебора всех возможных маршрутов

*import* itertools

Матрица стоимостей:

matrix = [  
 [330, 321, 432, 542, 500, 121, 420],  
 [234, 219, 673, 451, 344, 762, 424],  
 [456, 776, 443, 622, 632, 511, 299],  
 [107, 406, 140, 132, 342, 601, 261],  
 [632, 112, 532, 762, 544, 226, 331],  
 [633, 422, 773, 533, 721, 652, 117],  
 [438, 732, 645, 761, 304, 631, 760]]

Начальные параметры:

min\_cost = 100000000000000000000000  
min\_route = ''

Алгоритм:

*for* route *in* itertools.*permutations*(list(range(7))):  
 cost = 0  
 *for* i *in* range(len(route) - 1):  
 cost += matrix[route[i]][route[i + 1]]  
 cost += matrix[route[-1]][route[0]]  
 *if* cost < min\_cost:  
 min\_cost = cost  
 min\_route = ''.*join*([str(i + 1) *for* i *in* route])

Результат:

print("Минимальная стоимость маршрута:", min\_cost)  
print("Маршрут с минимальной стоимостью:", min\_route)





Решение верное, т.к. ответы сошлись.