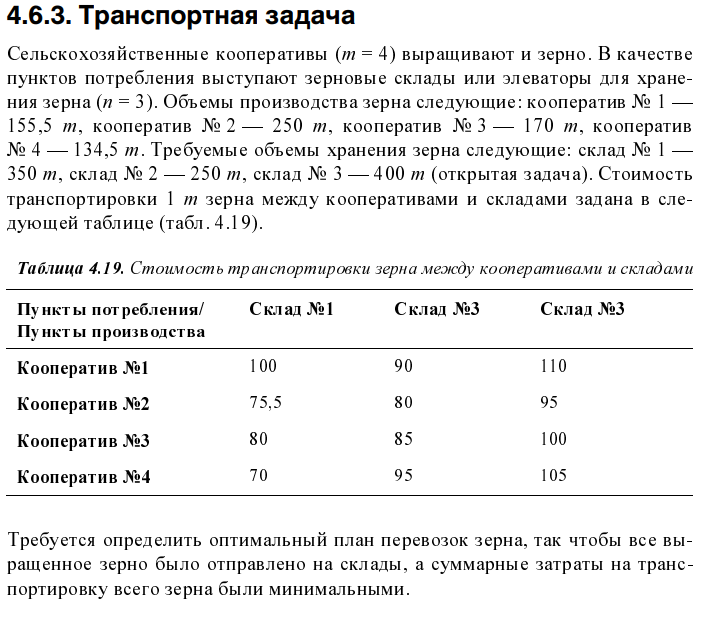
# Лабораторная работа №2

Студент: Болдинов Алексей ЭФМО-02-24

**Пример 1**.

Постановка задачи: см. файл Решение задач оптимизации в среде MS Excel (Александр Леоненков) стр. 201 (по учебнику).



Решение на Питоне:

from cvxopt.modeling import variable, op  
import time  
  
start = time.time()  
x = variable(12, 'x')  
c = [100, 90, 110, 75.5, 80, 95, 80, 85, 100, 70, 95, 105]  
z = (c[0] \* x[0] + c[1] \* x[1] + c[2] \* x[2] + c[3] \* x[3] + c[4] \* x[4] + c[5] \* x[5] + c[6] \* x[6] + c[7] \* x[7] + c[  
 8] \* x[8] + c[9] \* x[9] + c[10] \* x[10] + c[11] \* x[11])  
mass1 = (x[0] + x[1] + x[2] == 155.5)  
mass2 = (x[3] + x[4] + x[5] == 250)  
mass3 = (x[6] + x[7] + x[8] == 170)  
mass4 = (x[9] + x[10] + x[11] == 134.5)  
mass5 = (x[0] + x[3] + x[6] + x[9] <= 350)  
mass6 = (x[1] + x[4] + x[7] + x[10] <= 250)  
mass7 = (x[2] + x[5] + x[8] + x[11] <= 400)  
x\_non\_negative = (x >= 0)  
problem = op(z, [mass1, mass2, mass3, mass4, mass5, mass6, mass7, x\_non\_negative])  
problem.solve(solver='glpk')  
print("Результат Xopt:")  
  
for i in x.value:  
 print(i)  
print("Стоимость доставки:")  
print(problem.objective.value()[0])  
stop = time.time()  
print("Время :")  
print(stop - start)

Ответ программы:

GLPK Simplex Optimizer 5.0

19 rows, 12 columns, 36 non-zeros

0: obj = 0.000000000e+00 inf = 7.100e+02 (4)

6: obj = 6.300225000e+04 inf = 0.000e+00 (0)

\* 13: obj = 5.845525000e+04 inf = 0.000e+00 (0)

OPTIMAL LP SOLUTION FOUND

Результат Xopt:

0.0

155.5

0.0

45.5

94.5

110.0

170.0

0.0

0.0

134.5

0.0

0.0

Стоимость доставки:

58455.25

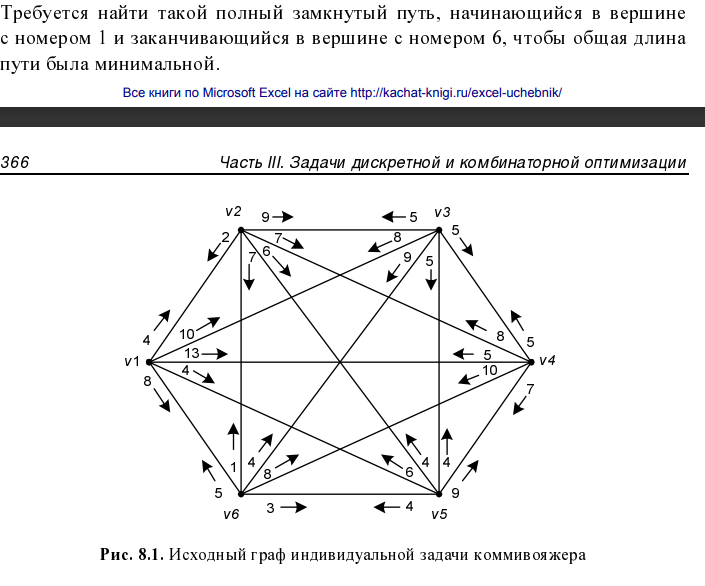
Время :

0.008278131484985352

Process finished with exit code 0

## Пример 2.1

Постановка задачи: см. файл Решение задач оптимизации в среде MS Excel (Александр Леоненков) стр. 366 (по учебнику).



Решение на Питоне:

import numpy as np  
from python\_tsp.exact import solve\_tsp\_dynamic\_programming  
  
distance\_matrix = np.array([  
 [0, 4, 10, 13, 4, 8],  
 [2, 0, 9, 7, 6, 7],  
 [8, 5, 0, 5, 5, 9],  
 [5, 8, 5, 0, 7, 10],  
 [6, 4, 4, 9, 0, 4],  
 [5, 1, 4, 8, 3, 0]  
])  
permutation, distance = solve\_tsp\_dynamic\_programming(distance\_matrix)  
print(permutation, distance)

Ответ программы:[0, 4, 2, 3, 5, 1] 26

## Пример 2.2

Постановка задачи: см. файл Решение задач оптимизации в среде MS Excel (Александр Леоненков) стр. 395 (по учебнику).



Решение на Питоне:

import numpy as np  
from python\_tsp.exact import solve\_tsp\_dynamic\_programming  
  
distance\_matrix = np.array([  
 [0, 15, 7, 10, 9, 21, 5, 11],  
 [17, 0, 10, 15, 7, 12, 6, 9],  
 [11, 9, 0, 13, 25, 14, 8, 10],  
 [12, 7, 13, 0, 21, 24, 10, 17],  
 [23, 8, 9, 13, 0, 15, 21, 16],  
 [17, 21, 8, 11, 13, 0, 10, 14],  
 [9, 11, 20, 15, 10, 17, 0, 8],  
 [7, 12, 17, 10, 9, 11, 22, 0]  
])  
permutation, distance = solve\_tsp\_dynamic\_programming(distance\_matrix)  
print(permutation, distance)

Ответ программы: [0, 6, 7, 5, 3, 1, 4, 2] 69