$L = 2x_1 + 3x_2 - x_4 \to max,$
$2x_1 - x_2 - 2x_4 + x_5 = 16,$
$3x_1 + 2x_2 + x_3 - 3x_4 = 18,$
$-x_1 + 3x_2 + 4x_4 + x_6 = 24,$
$x_1 \dots x_6 \geq 0;$

$L_{max} = 25.6$
$x_1 = 0.54, x_2 = 8.18, x_3 = 0, x_4 = 0, x_5 = 23.09, x_6 = 0$

	БП	x1	x2	x3	x4	x5	x6	СЧ	соотношения
17	f	-2	-3	0	1	0	0	0	
18	x5	2	-1	0	0	1	0	16,00	-16
19	x3	3	2	1	-3	0	0	18	9
20	x6	-1	3	0	4	0	1	24	8

	БП	x1	x2	x3	x4	x5	x6	СЧ	соотношения
24	f	-3	0	0	5	0	1	24	
25	x5	1,66666667	0	0	1,33333333	1	0,33333333	24	14,4
26	x3	3,66666667	0	1	-5,6666667	0	-0,66666667	2	0,545454545
27	x2	-0,3333333	1	0	1,33333333	0	0,33333333	8	-24

	БП	x1	x6	x3	x4	x5	x6	СЧ	соотношения
31	f	0	0	0,81818182	0,36363636	0	0,45454545	25,6363636	
32	x5	0	0	-0,4545455	3,90909091	1	0,63636364	23,0909091	
33	x1	1	0	0,27272727	-1,5454545	0	-0,18181818	0,54545455	
34	x2	0	1	0,09090909	0,81818182	0	0,27272727	8,18181818	

	БП	x1	x2	x3	x4	x5	x6	СЧ	соотношения
17	f	-2	-3	0	1	0	0	0	
18	x5	2	-1	0	0	1	0	16	=H19/C19
19	x3	3	2	1	-3	0	0	18	=H20/C20
20	x6	-1	3	0	4	0	1	24	=H21/C21

	БП	x1	x2	x3	x4	x5	x6	СЧ	соотношения
24	f	=B18-$C18/$C$21*B$21	=C18-$C18/$C$21*C$21	=D18-$C18/$C$21*D$21	=E18-$C18/$C$21*E$21	=F18-$C18/$C$21*F$21	=G18-$C18/$C$21*G$21	=H18-$C18/$C$21*H$21	
25	x5	=B19-$C19/$C$21*B$21	=C19-$C19/$C$21*C$21	=D19-$C19/$C$21*D$21	=E19-$C19/$C$21*E$21	=F19-$C19/$C$21*F$21	=G19-$C19/$C$21*G$21	=H19-$C19/$C$21*H$21	=H26/B26
26	x3	=B20-$C20/$C$21*B$21	=C20-$C20/$C$21*C$21	=D20-$C20/$C$21*D$21	=E20-$C20/$C$21*E$21	=F20-$C20/$C$21*F$21	=G20-$C20/$C$21*G$21	=H20-$C20/$C$21*H$21	=H27/B27
27	x2	=B21/$C$21	=C21/$C$21	=D21/$C$21	=E21/$C$21	=F21/$C$21	=G21/$C$21	=H21/$C$21	=H28/B28

	БП	x1	x6	x3	x4	x5	x6	СЧ	соотношения
31	f	=B25-$B25/$B$27*B$27	=C25-$B25/$B$27*C$27	=D25-$B25/$B$27*D$27	=E25-$B25/$B$27*E$27	=F25-$B25/$B$27*F$27	=G25-$B25/$B$27*G$27	=H25-$B25/$B$27*H$27	
32	x5	=B26-$B26/$B$27*B$27	=C26-$B26/$B$27*C$27	=D26-$B26/$B$27*D$27	=E26-$B26/$B$27*E$27	=F26-$B26/$B$27*F$27	=G26-$B26/$B$27*G$27	=H26-$B26/$B$27*H$27	
33	x1	=B27/$B$27	=C27/$B$27	=D27/$B$27	=E27/$B$27	=F27/$B$27	=G27/$B$27	=H27/$B$27	
34	x2	=B28-$B28/$B$27*B$27	=C28-$B28/$B$27*C$27	=D28-$B28/$B$27*D$27	=E28-$B28/$B$27*E$27	=F28-$B28/$B$27*F$27	=G28-$B28/$B$27*G$27	=H28-$B28/$B$27*H$27	

In [1]:
```
from scipy.optimize import linprog
import numpy as np
```

In [39]:
```
import warnings
warnings.filterwarnings("ignore")
```

In [20]:
```
def _rels(x: float, y: float):
    """Расчет отношений"""
    if y > 0:
        return x / y
    else:
        return np.inf

def simplex(A: list[list[float]], B: list[float], C: list[float], goal=min)
  """
    Реализация упрощенного симплекс-метода, предполагается, что исходная зад
    подается в каноническом виде и целевая функция выражена через свободные
    а также предполагается, что целевая функция ВСЕГДА минимизируется
    :param A list[list[float]]: Матрица коэффициентов ограничений
    :param B list[float]: Вектор правых частей ограничений
    :param C list[float]: Вектор коэффициентов целевой функции

    :return tuple: Функция возвращает кортеж из значений управляемых перемен
    и оптимальное значение функции

  """
  rels_vectorized = np.vectorize(_rels)

  A1 = np.column_stack((np.array(A), np.array(B))).astype(float)
  C1 = np.array(C + [0]).astype(float)
  if goal == "min":
    C1 *= -1

  basics = []
  # Выделение базисных элементов из условия задачи, предполагается, что
  # коэфициент при базисных переменных равен 1
  for i in range(A1.shape[0]):
    basics += [np.where((C1 == 0) * (A1[i, :] == 1))[0][0]]

  while (C1 > 0).any():
    basic = np.where(C1[:-1] == np.max(C1[:-1]))[0][0]
    coefs = A1[:, basic]
    t = rels_vectorized(A1[:, -1], coefs)
    free = np.where(t == np.min(t))[0][0]
    basics[free] = basic

    A1[free, :] /= A1[free, basic]


    for i in range(A1.shape[0]):
      if i != free:
        A1[i, :] -= A1[free, :] * A1[i, basic]

    C1 -= C1[basic] * A1[free, :]

  x = np.zeros(C1.shape[0] - 1)
  for i in range(len(basics)):
    x[basics[i]] = A1[i, -1]

  return x, C1[-1]
```

Задача из ДЗ

$$L = 2x_1 + 3x_2 - x_4 -> max$$

$$2x_1 - x2 - 2x_4 + x_5 = 16$$

$$3x_1 + 2x2 + x_3 - 3x_4 = 18$$

$$-x_1 + 3x2 + 4x_4 + x_6 = 24$$

$$x_1, \dots, x_6 >= 0$$

Данные

In [42]:
```
A = [[2, -1, 0, -2, 1, 0],
     [3, 2, 1, -3, 0, 0],
     [-1, 3, 0, 4, 0, 1]]

B = [16, 18, 24]

C = [-2, -3, 0, 1, 0, 0] # умножили коэфициенты на -1, что задачу на max об
```

scipy.optimize

In [43]:
```
solve = linprog(c=C, A_eq=A, b_eq=B, bounds=[(0, float("inf"))]*6, method='
print(solve.x)
print(solve.fun * (-1))
```

```
[ 0.54545455  8.18181818  0.          0.         23.09090909  0.        ]
-25.636363636363637
```

Собственная реализация

In [24]:
```
simplex(A, B, C, goal="min")
```

Out[24]:
```
(array([ 0.54545455,  8.18181818,  0.        ,  0.        , 23.09090909,
         0.        ]),
 -25.636363636363637)
```

Задача с лекции

$$L = 3x_1 + 4x_2 + 6x_3 -> max$$

$$2x_1 + 5x_2 + 2x_3 <= 12$$

$$7x_1 + x_2 + 2x_3 <= 18$$

$$x_1, x_2, x_3 >= 0$$

Данные

In [30]:
```
A = [[2, 5, 2, 1, 0],
     [7, 1, 2, 0, 1]]

B = [12, 18]

C = [-3, -4, -6, 0, 0]
```

scipy.optimize

In [40]:
```
solve = linprog(c=C, A_eq=A, b_eq=B, bounds=[(0, float("inf"))]*5, method='
print(solve.x)
print(solve.fun)
```

```
[0. 0. 6. 0. 6.]
-36.0
```

Собственная реализация

In [34]:
```
simplex(A, B, C, goal='min')
```

Out[34]:
```
(array([0., 0., 6., 0., 6.]), -36.0)
```