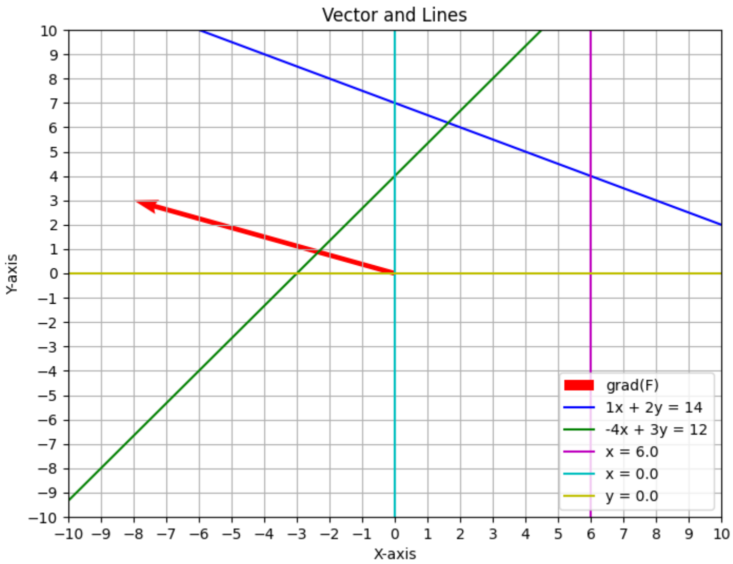
a

Работа №1

Графики были построены с помощью `matplotlib` в Python.

a) $F = -8x_1 + 3x_2 \to \min(\max)$

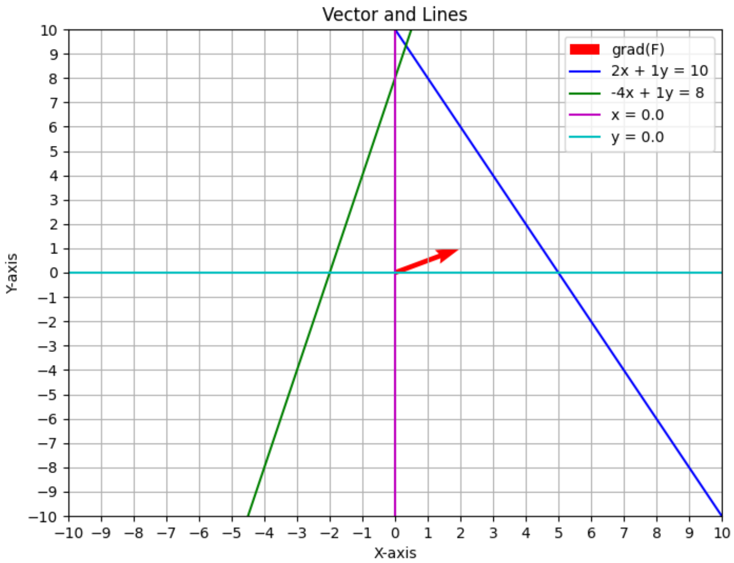
$$\begin{cases} x_1 + 2x_2 \leq 14 \\ -4x_1 + 3x_2 \leq 12 \\ x_1 \leq 6 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$



$$F_{\min} = F(6,0) = -48, \quad F_{\max} = F(0,4) = 12$$

b) $F = 2x_1 + x_2 \to \min(\max)$

$$\begin{cases} 2x_1 + x_2 \geq 10 \\ -4x_1 + x_2 \leq 8 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$

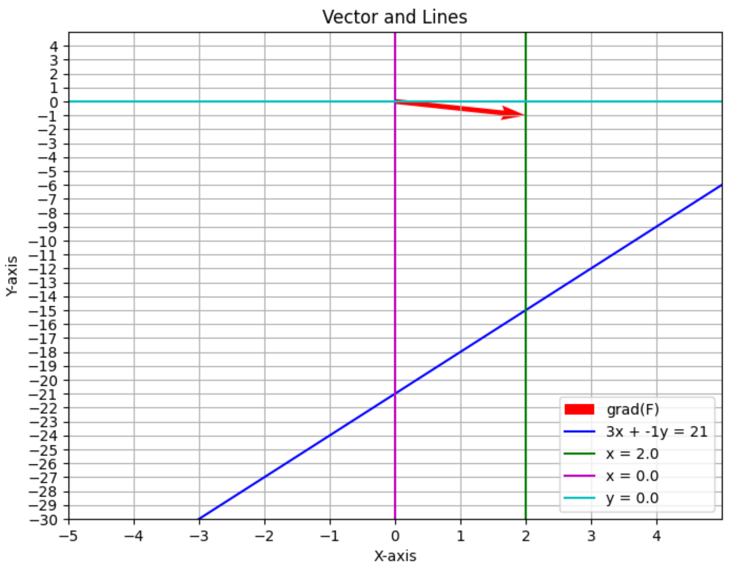


$F_{\min} = F(5,0) = 10$ (любая точка прямой $2x + y = 10$ подойдет, так как ∇F перпендикулярен ей)

$$\nexists F_{\max}$$

c) $F = 2x_1 - x_2 \to \min(\max)$

$$\begin{cases} 3x_1 - x_2 \geq 21 \\ x_1 \leq 2 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$



$$\nexists F_{\min}, \quad \nexists F_{\max}, \quad \text{у системы вообще нет решений}$$

Работа №2

$$F_{\min} = -(-F)_{\max} = -F'_{\max}$$

а) $F = -8x_1 + 3x_2 \to \min(\max)$

$$\begin{cases} x_1 + 2x_2 \le 14 \\ -4x_1 + 3x_2 \le 12 \\ x_1 \le 6 \\ x_1 \ge 0, \quad x_2 \ge 0 \end{cases}$$

$$\begin{cases} x_1 + 2x_2 + x_3 = 14 \\ -4x_1 + 3x_2 + x_4 = 12 \\ x_1 + x_5 = 6 \\ x_i \ge 0, i = \overline{1,5} \end{cases}$$

	c	-8	3	0	0	0	0	
Basis	C base	x0	x1	x2	x3	x4	B	reduced_cost
A2	0	1	2	1	0	0	14	7
A3	0	-4	3	0	1	0	12	4
A4	0	1	0	0	0	1	6	∞
	delta	-8	3	0	0	0	0	

	c	-8	3	0	0	0	0
Basis	C base	x0	x1	x2	x3	x4	B
A2	0	3.66667	0	1	-0.666667	0	6
A1	3.0	-1.33333	1	0	0.333333	0	4
A4	0	1	0	0	0	1	6
	delta	-4	0	0	-1	0	12

$$x_1 = 14 - 2x_2 - x_3 = 14 - 2 \cdot 4 - 6 = 0, \quad F_{\max} = F(0,4) = 12$$

Basis	C base	x0	x1	x2	x3	x4	B	reduced_cost
A2	0	1	2	1	0	0	14	14
A3	0	-4	3	0	1	0	12	-3
A4	0	1	0	0	0	1	6	6
	delta	8	-3	0	0	0	0	
	c	8	-3	0	0	0	0	

Таблица 1: Таблица с данными и столбцом reduced_cost.

Basis	C base	x0	x1	x2	x3	x4	B
A2	0	0	2	1	0	-1	8
A3	0	0	3	0	1	4	36
A0	8.0	1	0	0	0	1	6
	delta	0	-3	0	0	-8	48
	c	8	-3	0	0	0	0

Таблица 2: Таблица с данными без столбца reduced_cost.

$$x_1 + x_3 = 14 \quad \Rightarrow \quad x_2 = 0$$

$$F_{\min} = F(6, 0) = -48$$

b) $F = 2x_1 + x_2 \to \min(\max)$

$$\begin{cases} 2x_1 + x_2 \geq 10 \\ -4x_1 + x_2 \leq 8 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$

$$\begin{cases} -2x_1 - x_2 + x_3 = -10 \\ -4x_1 + x_2 + x_4 = 8 \\ x_i \geq 0, i = \overline{1,4} \end{cases}$$

Basis	C base	x0	x1	x2	x3	B	reduced_cost
A2	0	-2	-1	1	0	-10	5
A3	0	-4	1	0	1	8	-2
	delta	2	1	0	0	0	
	c	2	1	0	0	0	

$$\exists \delta_j > 0 : \forall x_{ij} < 0 \quad \Rightarrow \quad \nexists F_{\max}$$

Basis	C base	x0	x1	x2	x3	B
A2	0	-2	-1	1	0	-10
A3	0	-4	1	0	1	8
	delta	-2	-1	0	0	0

Basis	C base	x0	x1	x2	x3	B
A0	-2	1	0.5	-0.5	0	5
A3	0	0	3	-2	1	28
	delta	0	0	-1	0	10

$\Rightarrow$ Задача имеет бесконечное множество оптимальных решений

$$F_{\min} = F(5, 0) = 10$$

c) $F = 2x_1 - x_2 \to \min(\max)$

$$\begin{cases} 3x_1 - x_2 \geq 21 \\ x_1 \leq 2 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$

$$\begin{cases} 3x_1 - x_2 - x_3 = 21 \\ x_1 + x_4 = 2 \\ x_i \geq 0, i = \overline{1,4} \end{cases}$$

Basis	C base	x1	x2	x3	x4	B	r
A3	0	-3	1	1	0	-21	7
A4	0	1	0	0	1	2	2
	delta	2	-1	0	0	0	

Basis	C base	x1	x2	x3	x4	B
A3	0	0	1	1	3	-15
A1	2	1	0	0	1	2
	delta	0	-1	0	-2	-4

$$(\forall j : \delta_j \leq 0) \quad \& \quad F < 0 \quad \Rightarrow \quad D = \emptyset$$

$$\nexists F_{\min}, \quad \nexists F_{\max}$$

Работа №3

а) $F = -8x_1 + 3x_2 \to \min(\max)$

$$\begin{cases} x_1 + 2x_2 \le 14 \\ -4x_1 + 3x_2 \le 12 \\ x_1 \le 6 \\ x_1 \ge 0, \quad x_2 \ge 0 \end{cases}$$

Решим сначала на максимум:

Решение исходной задачи:

$$x_{\max} = (0, 4)$$

Формулировка двойственной задачи:

1. **Целевая функция:**

$$F^* = 14y_1 + 12y_2 + 6y_3 \to \min$$

2. **Ограничения:**

$$\begin{cases} y_1 - 4y_2 + y_3 \ge -8 \\ 2y_1 + 3y_2 \ge 3 \\ y_1 \ge 0, \quad y_2 \ge 0, \quad y_3 \ge 0 \end{cases}$$

Решение:

$$\begin{cases} (x_1 + 2x_2 - 14)y_1^* = 0 \\ (-4x_1 + 3x_2 - 12)y_2^* = 0 \\ (x_1 - 6)y_3^* = 0 \end{cases}$$

$$\begin{cases} (0 + 2 \cdot 4 - 14)y_1^* = 0 \Rightarrow y_1^* = 0 \\ (-4 \cdot 0 + 3 \cdot 4 - 12)y_2^* = 0 \Rightarrow y_2^* > 0 \\ (0 - 6)y_3^* = 0 \Rightarrow y_3^* = 0 \end{cases}$$

Теорема о дополняющей нежесткости:

Если x_1 и x_2 — оптимальное решение прямой задачи, а y_1, y_2, y_3 — оптимальное решение двойственной задачи, то:

1. $x_1(y_1 - 4y_2 + y_3 + 8) = 0$ 2. $x_2(2y_1 + 3y_2 - 3) = 0$

Из решения прямой задачи:

$$\begin{cases} 0 \cdot (y_1 - 4y_2 + y_3 + 8) = 0 \\ 4 \cdot (2y_1 + 3y_2 - 3) = 0 \end{cases} \Rightarrow y^* = (0, 1, 0) \text{ — Оптимальное решение двойственной задачи}$$

$$F^* = 14 \cdot 0 + 12 \cdot 1 + 6 \cdot 0 = 12$$

Решим на максимум через теорему 3:

Берем данные из последней таблицы решения этого номера в работе 2:

$$y^* = \begin{pmatrix} 0 & 3 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & -0.67 & 0 \\ 0 & 0.33 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (0, 1, 0)$$

Получили то же самое.

Решим теперь на минимум. Решим так же, находя максимум функции $G = -F$:

Решение исходной задачи:

$$x_{\max} = (6, 0)$$

Прямая задача:

$$G = 8x_1 - 3x_2 \to \max$$

$$\begin{cases} x_1 + 2x_2 \leq 14 \\ -4x_1 + 3x_2 \leq 12 \\ x_1 \leq 6 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$

Двойственная задача:

1. **Целевая функция:**

$$G^* = 14y_1 + 12y_2 + 6y_3 \to \max$$

2. **Ограничения:**

$$\begin{cases} y_1 - 4y_2 + y_3 \geq 8 \\ 2y_1 + 3y_2 \geq -3 \\ y_1 \geq 0, \quad y_2 \geq 0, \quad y_3 \geq 0 \end{cases}$$

Решение:

$$\begin{cases} (x_1 + 2x_2 - 14)y_1^* = 0 \\ (-4x_1 + 3x_2 - 12)y_2^* = 0 \\ (x_1 - 6)y_3^* = 0 \end{cases}$$

$$\begin{cases} (6 + 2 \cdot 0 - 14)y_1^* = 0 \Rightarrow y_1^* = 0 \\ (-4 \cdot 6 + 3 \cdot 0 - 12)y_2^* = 0 \Rightarrow y_2^* = 0 \\ (6 - 6)y_3^* = 0 \Rightarrow y_3^* > 0 \end{cases}$$

Теорема о дополняющей нежесткости:

Если x_1 и x_2 — оптимальное решение прямой задачи, а y_1, y_2, y_3 — оптимальное решение двойственной задачи, то:

1. $x_1(y_1^* - 4y_2^* + y_3^* - 8) = 0$ 2. $x_2(2y_1^* + 3y_2^* + 3) = 0$

Из решения прямой задачи $x_1 = 6$ и $x_2 = 0$:

$$\begin{cases} 6 \cdot (y_1^* - 4y_2^* + y_3^* - 8) = 0 \\ 0 \cdot (2y_1^* + 3y_2^* + 3) = 0 \end{cases} \Rightarrow y^* = (0, 0, 8) \text{ — Оптимальное решение двойственной задачи}$$

$$G^* = 14 \cdot 0 + 12 \cdot 0 + 6 \cdot 8 = 48 \Rightarrow F^* = -G^* = -48$$

b) $F = 2x_1 + x_2 \to \min(\max)$

$$\begin{cases} 2x_1 + x_2 \geq 10 \\ -4x_1 + x_2 \leq 8 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$

$\nexists F_{\max}$

Решим теперь на минимум. Решим так же, находя максимум функции $G = -F$:

Решение исходной задачи:

$$x_{\max} = (5, 0)$$

Прямая задача:

$$G = -2x_1 - x_2 \to \max$$

$$\begin{cases} -2x_1 - x_2 \leq -10 \\ -4x_1 + x_2 \leq 8 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$

Двойственная задача:

1. **Целевая функция:**

$$G^* = -10y_1 + 8y_2 \to \max$$

2. **Ограничения:**

$$\begin{cases} -2y_1 - 4y_2 \geq -2 \\ -y_1 + y_2 \geq -1 \\ y_1 \geq 0, \quad y_2 \geq 0 \end{cases}$$

Решение:

$$\begin{cases} (-2x_1 - x_2 + 10)y_1^* = 0 \\ (-4x_1 + x_2 - 8)y_2^* = 0 \end{cases}$$

$$\begin{cases} (-2 \cdot 5 - 0 + 10)y_1^* = 0 \Rightarrow y_1^* > 0 \\ (-4 \cdot 5 + 0 - 8)y_2^* = 0 \Rightarrow y_2^* = 0 \end{cases}$$

Теорема о дополняющей нежесткости:

1. $x_1(-2y_1 - 4y_2 + 2) = 0$ 2. $x_2(-y_1 + y_2 + 1) = 0$
   Из решения прямой задачи:

$$\begin{cases} 5 \cdot (-2y_1 - 4y_2 + 2) = 0 \\ 0 \cdot (-y_1 + y_2 + 1) = 0 \end{cases} \Rightarrow y^* = (1, 0) \text{ — Оптимальное решение двойственной задачи}$$

$$G^* = -10 \cdot 1 + 8 \cdot 0 = -10 \Rightarrow F^* = -G^* = 10$$

c) $F = 2x_1 - x_2 \to \min(\max)$

$$\begin{cases} 3x_1 - x_2 \geq 21 \\ x_1 \leq 2 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$

$\nexists F_{\min}, \nexists F_{\max}$

Работа №4

Задание 3

Для задач № 3, 4 составить математическую модель для прямой и двойственной задачи. Получить решение прямой и двойственной задачи симплекс-методом. Дать экономическую интерпретацию двойственных задач и двойственных оценок.

Задача №3:

Для производства четырех видов изделий (А, В, С) предприятие использует три вида сырья: металл, пластмассу, резину. Запасы сырья, технологические коэффициенты (расход каждого вида сырья на производство единицы каждого изделия) представлены в таблице 2 (варианты 1…20). В ней же указана прибыль от реализации одного изделия каждого вида. Требуется составить такой план выпуска указанных изделий, чтобы обеспечить максимальную прибыль.

Вариант 9					
металл	2	1	3	1	2300
пластмасса	4	1	6	5	1500
резина	4	7	9	10	1000
Прибыль (руб)	8	4	2	1	

Рис. 1: Изображение постановки задачи

Постановка задачи

Прямая задача

Найти оптимальный план производства продукции с максимальной прибылью, для которого достаточно имеющихся ресурсов.

$$x_1, x_2, x_3, x_4 \text{ — количество произведенной продукции.}$$

Целевая функция:

$$F = 8x_1 + 4x_2 + 2x_3 + 1x_4 \to \max$$

Ограничения:

$$\begin{cases} 2x_1 + 1x_2 + 3x_3 + 1x_4 \leq 2300 \\ 4x_1 + 1x_2 + 6x_3 + 5x_4 \leq 1500 \\ 4x_1 + 7x_2 + 9x_3 + 10x_4 \leq 1000 \\ x_1, x_2, x_3, x_4 \geq 0 \end{cases}$$

Двойственная задача

Оценить каждый из видов сырья, используемого для производства продукции. Оценки, приписываемые каждому виду сырья, должны быть такими, чтобы оценка всего используемого сырья была минимальна, а суммарная оценка сырья, используемого для производства единицы продукции, — не меньше цены единицы продукции. **Целевая функция двойственной задачи**:

$$G = 2300y_1 + 1500y_2 + 1000y_3 \to \min$$

Ограничения:

$$\begin{cases} 2y_1 + 4y_2 + 4y_3 \geq 8 \\ 1y_1 + 1y_2 + 7y_3 \geq 4 \\ 3y_1 + 6y_2 + 9y_3 \geq 2 \\ 1y_1 + 5y_2 + 10y_3 \geq 1 \\ y_1, y_2, y_3 \geq 0 \end{cases}$$

Решим прямую задачу

$$F = 8x_1 + 4x_2 + 2x_3 + 1x_4 + 0x_5 + 0x_6 + 0x_7 \to \max$$

$$\begin{cases} 2x_1 + 1x_2 + 3x_3 + 1x_4 + 1x_5 + 0x_6 + 0x_7 = 2300 \\ 4x_1 + 1x_2 + 6x_3 + 5x_4 + 0x_5 + 1x_6 + 0x_7 = 1500 \\ 4x_1 + 7x_2 + 9x_3 + 10x_4 + 0x_5 + 0x_6 + 1x_7 = 1000 \\ x_1, x_2, x_3, x_4 \geq 0 \end{cases}$$

Basis	C base	x1	x2	x3	x4	x5	x6	x7	B	reduced_cost
A5	0	2	1	3	1	1	0	0	2300	1150
A6	0	4	1	6	5	0	1	0	1500	375
A7	0	4	7	9	10	0	0	1	1000	250
	delta	8	4	2	1	0	0	0	0	
	c	8	4	2	1	0	0	0	0	

Basis	C base	x1	x2	x3	x4	x5	x6	x7	B
A5	0	0	-2.5	-1.5	-4	1	0	-0.5	1800
A6	0	0	-6	-3	-5	0	1	-1	500
A1	8.0	1	1.75	2.25	2.5	0	0	0.25	250
	delta	0	-10	-16	-19	0	0	-2	2000
	c	8	4	2	1	0	0	0	0

$$x_1 = 250,\ x_5 = 1800,\ x_6 = 500$$

$$\begin{cases} 1x_2 + 3x_3 + 1x_4 = 0 \\ 1x_2 + 6x_3 + 5x_4 = 0 \\ 7x_2 + 9x_3 + 10x_4 + 1x_7 = 0 \end{cases}$$

$$\begin{cases} 1x_2 + 3x_3 + 1x_4 = 0 \\ 3x_3 + 4x_4 = 0 \\ 7x_2 + 9x_3 + 10x_4 + 1x_7 = 0 \end{cases}$$

$$\begin{cases} 1x_2 + 3x_3 + -\frac{3}{4}x_3 = 0 \\ x_4 = -\frac{3}{4}x_3 \\ 7x_2 + 9x_3 - \frac{30}{4}x_3 + 1x_7 = 0 \end{cases}$$

$$\begin{cases} 1x_2 + \frac{9}{4}x_3 = 0 \\ x_4 = -\frac{3}{4}x_3 \\ 7x_2 + \frac{6}{4}x_3 + 1x_7 = 0 \end{cases}$$

$$F_{\max} = 2000 = 8 \cdot 250 + 4x_2 + 2x_3 + x_4 = 2000 + 4x_2 + \frac{5}{4}x_3 \Rightarrow 4x_2 + \frac{5}{4}x_3 = 0$$

$$\begin{cases} 1x_2 + \frac{9}{4}x_3 = 0 \\ 4x_2 + \frac{5}{4}x_3 = 0 \\ 7x_2 + \frac{6}{4}x_3 + 1x_7 = 0 \\ x_4 = -\frac{3}{4}x_3 \end{cases}$$

$$\begin{cases} x_1 = 250 \\ x_5 = 1800 \\ x_6 = 500 \\ x_2 = x_3 = x_4 = x_7 = 0 \end{cases}$$

$$F_{\max} = 2000$$

Решим обратную задачу

$$y^* = \begin{pmatrix} 0 & 0 & 8 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & -0.5 \\ 0 & 1 & -1 \\ 0 & 0 & 0.25 \end{pmatrix} = (0, 0, 2)$$

$$G_{\min} = 2300 \cdot 0 + 1500 \cdot 0 + 1000 \cdot 2 = 2000 = F_{\max}$$

Видим, что первый и второй материал в избытке, третий материал дефицитный.

$$\begin{cases} 2 \cdot 0 + 4 \cdot 0 + 4 \cdot 2 = 8 \\ 1 \cdot 0 + 1 \cdot 0 + 7 \cdot 2 > 4 \\ 3 \cdot 0 + 6 \cdot 0 + 9 \cdot 2 > 2 \\ 1 \cdot 0 + 5 \cdot 0 + 10 \cdot 2 > 1 \end{cases}$$

Первое ограничение выполняется как равенство $\Rightarrow$ двойственные оценки ресурсов, используемых для производства единицы продукции A, равны в точности доходам $\Rightarrow$ производить это изделие целесообразно ($x_1^* = 250$).

Второе, третье и четвертое ограничения выполняются как больше $\Rightarrow$ производить изделия B, C и S экономически невыгодно

$$x_2^* = 0, x_3^* = 0, x_4^* = 0$$

Анализ устойчивости двойственных оценок

$$x_b^* = x_b + A_b^{-1} \cdot (b + \Delta b)$$

$$A_b^{-1} \cdot (b + \Delta b) =$$

$$\begin{pmatrix} 1 & 0 & -0.5 \\ 0 & 1 & -1 \\ 0 & 0 & 0.25 \end{pmatrix} \cdot \begin{pmatrix} 2300 + \Delta b_1 \\ 1500 + \Delta b_2 \\ 1000 + \Delta b_3 \end{pmatrix} =$$

$$\begin{pmatrix} 1800 + \Delta b_1 - 0.5\Delta b_3 \\ 500 + \Delta b_2 - \Delta b_3 \\ 250 + 0.25\Delta b_3 \end{pmatrix} \geq 0$$

$$1) \Delta b_2 = \Delta b_3 = 0 \Rightarrow \Delta b_1 \in [-1800, +\infty)$$

$$2) \Delta b_1 = \Delta b_3 = 0 \Rightarrow \Delta b_2 \in [-500, +\infty)$$

$$3) \Delta b_1 = \Delta b_2 = 0 \Rightarrow \Delta b_3 \in [-1000, 500]$$

Предположим: $\Delta b_2 = -100; \Delta b_3 = -200$

$$\begin{pmatrix} 1800 - 0.5 \cdot (-200) \\ 500 - (-200) \\ 250 + 0.25 \cdot (-200) \end{pmatrix} = \begin{pmatrix} 1900 \\ 700 \\ 200 \end{pmatrix} \geq 0$$

$$x_b^* = \begin{pmatrix} 1800 + 1900 \\ 500 + 700 \\ 250 + 200 \end{pmatrix} = \begin{pmatrix} 3700 \\ 1200 \\ 450 \end{pmatrix}$$

$$F^* = 8 \cdot 3700 + 4 \cdot 1200 + 2 \cdot 450 + 1 \cdot 450 = 5750$$

Видим, что прибыль увеличилась.

Работа №5

Транспортная задача

Требуется найти такой план перевозок продукции от поставщиков к потребителям, который обеспечивал бы спрос потребителей и вывоз продукции от поставщиков при минимальных суммарных транспортных расходах.

	B1	B2	B3	B4	B5	Запасы
A1	3	3	5	3	3	150
A2	7	3	6	1	3	50
A3	2	8	7	2	9	100
A4	1	3	9	6	4	100
Потребности	50	150	50	100	50	

Таблица 3: Исходные данные транспортной задачи

Нахождение первого опорного решения

$$\sum \text{потребностей} = 400$$

$$\sum \text{запасов} = 400 = \sum \text{потребностей} \Rightarrow$$

Задача является задачей с правильным балансом.

Метод северо-западного угла

	B1	B2	B3	B4	B5	Запасы
A1	3 50	3 100	5	3	3	150
A2	7	3 50	6	1	3	50
A3	2	8 0	7 50	2 50	9	100
A4	1	3	9	6 50	4 50	100
Потребности	50	150	50	100	50	

Таблица 4: Опорное решение методом северо-западного угла

Получили решение:

$$F = 50 \cdot 3 + 100 \cdot 3 + 50 \cdot 3 + 0 \cdot 8 + 50 \cdot 7 + 50 \cdot 2 + 50 \cdot 6 + 50 \cdot 4 = 1550$$

Метод минимальных элементов

	B1	B2	B3	B4	B5	Запасы
A1	3	3 100	5	3	3 50	150
A2	7	3 0	6	1 50	3	50
A3	2	8	7 50	2 50	9	100
A4	1 50	3 50	9	6	4	100
Потребности	50	150	50	100	50	

Таблица 5: Опорное решение методом минимальных элементов

Получили решение:

$$F = 100 \cdot 3 + 50 \cdot 3 + 50 \cdot 1 + 0 \cdot 8 + 50 \cdot 7 + 50 \cdot 2 + 50 \cdot 1 + 50 \cdot 3 = 1150$$

Метод потенциалов

Возьмем опорное решение, полученное методом минимального элемента

	1	3	6	1	3	Запасы
0	3	3 100	5 p=-1	3	3 50	150
0	7	3 0	6	1 50	3	50
1	2	8	7 50	2 50	9	100
0	1 50	3 50	9	6	4	100
Потребности	50	150	50	100	50	

Таблица 6: Первый шаг метода потенциалов

	1	3	5	0	3	Запасы
0	3	3 50	5 50	3	3 50	150
0	7	3 50	6	1	3	50
2	2 p=-1	8	7 0	2 100	9	100
0	1 50	3 50	9	6	4	100
Потребности	50	150	50	100	50	

Таблица 7: Второй шаг метода потенциалов

	1	3	5	1	3	Запасы
0	3	3 50	5 50	3	3 50	150
0	7	3 50	6	1	3	50
1	2 0	8	7	2 100	9	100
0	1 50	3 50	9	6	4	100
Потребности	50	150	50	100	50	

Таблица 8: Третий шаг метода потенциалов

Теперь все $\Delta_{i,j} = c_{i,j} - u_i - v_j \geq 0 \Rightarrow$ найдено оптимальное решение.
Ответ:

$$X = \begin{pmatrix} 0 & 50 & 50 & 0 & 50 \\ 0 & 50 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 \\ 50 & 50 & 0 & 0 & 0 \end{pmatrix}$$

$$F_{\min} = 3 \cdot 50 + 5 \cdot 50 + 3 \cdot 50 + 3 \cdot 50 + 2 \cdot 100 + 1 \cdot 50 + 3 \cdot 50 = 1100$$

Решение с помощью кода

```
from pulp import LpProblem, LpMinimize, LpVariable, lpSum, LpStatus
suppliers = ['A1', 'A2', 'A3', 'A4']
customers = ['B1', 'B2', 'B3', 'B4', 'B5']
  supply = {
    'A1': 150,
    'A2': 50,
    'A3': 100,
    'A4': 100
}
demand = {
    'B1': 50,
    'B2': 150,
```

```
    'B3': 50,
    'B4': 100,
    'B5': 50
}
costs = {
    ('A1', 'B1'): 3, ('A1', 'B2'): 3, ('A1', 'B3'): 5, ('A1', 'B4'): 3, ('A1', 'B5'): 3,
    ('A2', 'B1'): 7, ('A2', 'B2'): 3, ('A2', 'B3'): 6, ('A2', 'B4'): 1, ('A2', 'B5'): 3,
    ('A3', 'B1'): 2, ('A3', 'B2'): 8, ('A3', 'B3'): 7, ('A3', 'B4'): 2, ('A3', 'B5'): 9,
    ('A4', 'B1'): 1, ('A4', 'B2'): 3, ('A4', 'B3'): 9, ('A4', 'B4'): 6, ('A4', 'B5'): 4
}
prob = LpProblem("Transportation_Problem", LpMinimize)

# Decision variables: x[supplier][customer]
x = LpVariable.dicts("Shipments", [(s, c) for s in suppliers for c in customers], lowBound=0, cat='C

# Objective function: minimize total transportation cost
prob += lpSum([costs[(s, c)] * x[(s, c)] for s in suppliers for c in customers]), "Total_Cost"

# Supply constraints
for s in suppliers:
    prob += lpSum([x[(s, c)] for c in customers]) <= supply[s], f"Supply_{s}"

# Demand constraints
for c in customers:
    prob += lpSum([x[(s, c)] for s in suppliers]) >= demand[c], f"Demand_{c}"

# Solve the problem
prob.solve()

# Check the status of the solution
print("Status:", LpStatus[prob.status])

# Print the optimal shipment amounts and total cost
print("Optimal Shipments:")
for s in suppliers:
    for c in customers:
        if x[(s, c)].varValue > 0:
            print(f"{s} -> {c}: {x[(s, c)].varValue}")

print(f"\nTotal Minimum Cost: {prob.objective.value()}")
```

Вывод:

```
Status: Optimal
Optimal Shipments:
A1 -> B2: 50.0
A1 -> B3: 50.0
A1 -> B5: 50.0
A2 -> B2: 50.0
A3 -> B4: 100.0
A4 -> B1: 50.0
A4 -> B2: 50.0

Total Minimum Cost: 1100.0
```

Работа №6

Задача

$$F = 2x_1 + 1x_2 \to \min(\max)$$

$$\begin{cases} 4x_1 + x_2 \leq 16 \\ x_1 + x_2 \leq 5 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$

1. Решение задачи геометрическим методом

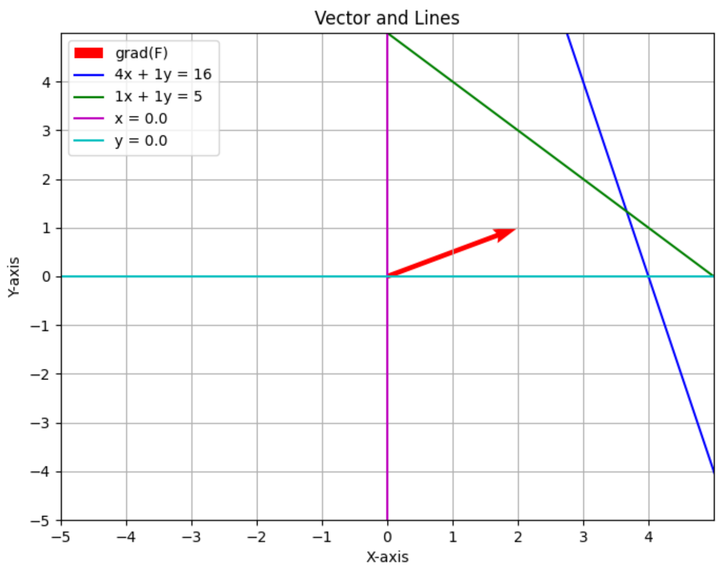


Рис. 2: Графическое решение задачи

$$F_{\min} = F(0,0) = 0, \quad F_{\max} = F\left(\frac{11}{3}, \frac{4}{3}\right) = \frac{26}{3}$$

2. Решение задачи симплекс-методом

$$F_{\min} = -(-F)_{\max} = -F'_{\max}$$

$$\begin{cases} x_1 + 2x_2 + x_3 = 14 \\ -4x_1 + 3x_2 + x_4 = 12 \\ x_1 + x_5 = 6 \\ x_i \geq 0, i = \overline{1,5} \end{cases}$$

Basis	C base	x0	x1	x2	x3	B	reduced_cost
A2	0	4	1	1	0	16	4
A3	0	1	1	0	1	5	5
	delta	2	1	0	0	0	
	c	2	1	0	0	0	

Basis	C base	x0	x1	x2	x3	B	reduced_cost
A0	2.0	1	0.25	0.25	0	4	16
A3	0	0	0.75	-0.25	1	1	1.33333
	delta	0	0.5	-0.5	0	8	
	c	2	1	0	0	0	

Basis	C base	x0	x1	x2	x3	B
A0	2.0	1	0	0.333333	-0.333333	3.66667
A1	1.0	0	1	-0.333333	1.33333	1.33333
	delta	0	0	-0.333333	-0.666667	8.66667
	c	2	1	0	0	0

$$F_{\max} = F\left(\frac{11}{3}, \frac{4}{3}\right) = \frac{26}{3}$$

Basis	C base	x0	x1	x2	x3	B
A2	0	4	1	1	0	16
A3	0	1	1	0	1	5
	delta	-2	-1	0	0	0
	c	-2	-1	0	0	0

$$F_{\min} = F(0,0) = 0$$

3. Решение задачи методом отсечения Гомори

Добавляем неравенство:

$$\frac{2}{3} - \left(\frac{1}{3} \cdot x_2 + \frac{2}{3} \cdot x_3\right) \leq 0$$

3.1 Геометрическим методом

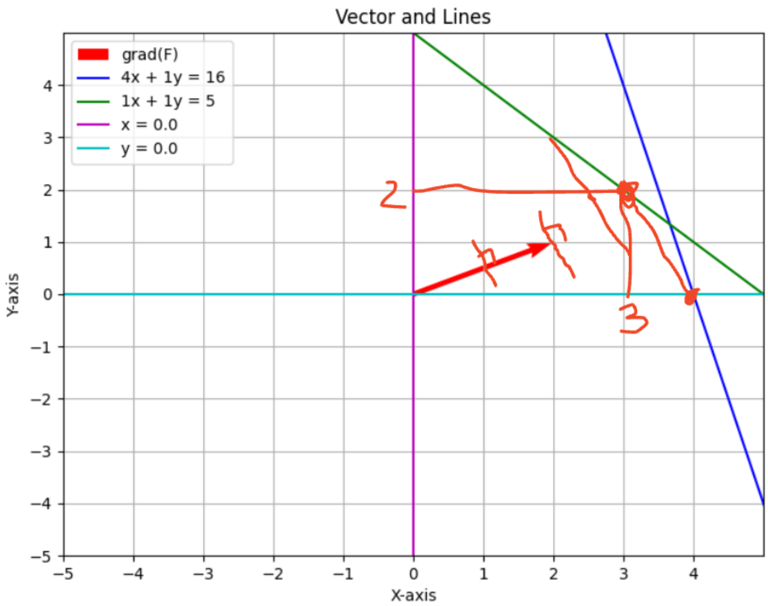


Рис. 3: Графическое решение задачи с отсечением Гомори

$$F_{\max} = F(3,2) = F(4,0) = 8$$

3.2 Симплекс-методом

Базис	B	x1	x2	x3	x4	x5
x1	11/3	1	0	1/3	-1/3	0
x2	4/3	0	1	-1/3	4/3	0
x5	-2/3	0	0	-1/3	-2/3	1
F(X0)	-26/3	0	0	-1/3	-2/3	0

$$\theta_3 = -1/3 : (-1/3) = 1\theta_4 = -2/3 : (-2/3) = 1$$

Базис	B	x1	x2	x3	x4	x5
x1	4	1	0	1/2	0	-1/2
x2	0	0	1	-1	0	2
x4	1	0	0	1/2	1	-3/2
F(X0)	-8	0	0	0	0	-1

Получили, что $x_1 = 4$, $x_2 = 0$, $F(x_1, x_2) = 8$.

Работа №7

Метод ветвей и границ решения задачи коммвояжера

Дано n городов, $C = |c_{i,j}|, \quad i, j = \overline{1,n}$ - матрица стоимостей переездов из i-х городов в j-е. Коммивояжер должен выехать из своего города, заехать в каждый город только один раз и вернуться в исходный город. Нужно найти замкнутый маршрут объезда всех городов минимальной стоимости.

$$x_{i,j} = \begin{cases} 1, \text{если коммивояжер едет из i в j} \\ 0, \text{иначе} \end{cases} \qquad F = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{i.j} \cdot x_{i,j} \to min$$

$$\sum_{i=1}^{n} x_{i,j} = 1 \ \forall j = \overline{1,n}, \qquad \sum_{j=1}^{n} x_{i,j} = 1 \ \forall i = \overline{1,n}$$

Из каждого i-го города только один выезд, в каждый j-й город, только один въезд.

$u_i \in \overline{1,n}$- каким по счету мы посетим город i.

$u_i - u_j + n \cdot x_{i,j} \leq n - 1$ - обеспечивает замкнутость маршрута и отсутствие петель

0.0.1 Решение программой

```
import random

class TSPSolver:
    def __init__(self, matrix):
        self.n = len(matrix)
        self.graph = matrix
        self.best_cost = float('inf')
        self.best_path = []

    def solve(self):
        visited = [False] * self.n
        visited[0] = True
        self._branch_and_bound(0, visited, 1, 0, [0])
        return self.best_cost, self.best_path

    def _branch_and_bound(self, curr_node, visited, level, curr_cost, path):
        if level == self.n:
            return_cost = self.graph[curr_node][0]
            if return_cost > 0:
                total_cost = curr_cost + return_cost
                if total_cost < self.best_cost:
                    self.best_cost = total_cost
                    self.best_path = path + [0]
            return

        for i in range(self.n):
            if not visited[i] and self.graph[curr_node][i] > 0:
                next_cost = curr_cost + self.graph[curr_node][i]
                if next_cost < self.best_cost:
                    visited[i] = True
                    self._branch_and_bound(i, visited, level + 1, next_cost, path + [i])
                    visited[i] = False

def input_matrix(n):
    print(f"Введите матрицу размера {n} x {n} (через пробелы, элементы разделяются строками):")
    matrix = []
    for i in range(n):
        row = list(map(int, input().split()))
```

```
        matrix.append(row)
    return matrix

def generate_random_matrix(n, max_weight=100):
    matrix = [[random.randint(1, max_weight) if i != j else 0 for j in range(n)] for i in range(n)]
    return matrix

def print_matrix(matrix):
    for row in matrix:
        print(" ".join(map(str, row)))

def main():
    n = int(input("Введите размер матрицы (n): "))
    print("Выберите опцию:")
    print("1. Ввести свою матрицу")
    print("2. Сгенерировать случайную матрицу")
    option = int(input("Ваш выбор (1 или 2): "))

    if option == 1:
        matrix = input_matrix(n)
    elif option == 2:
        matrix = generate_random_matrix(n)
        print("Сгенерированная матрица:")
        print_matrix(matrix)
    else:
        print("Некорректный выбор.")
        return

    solver = TSPSolver(matrix)
    cost, path = solver.solve()
    print("\nМинимальная стоимость:", cost)
    print("Оптимальный путь:", " -> ".join(map(str, path)))

if __name__ == "__main__":
    main()
```

Работа программы:

```
Введите размер матрицы (n): 10
Выберите опцию:
1. Ввести свою матрицу
2. Сгенерировать случайную матрицу
Ваш выбор (1 или 2): 2
Сгенерированная матрица:
0 34 68 18 63 80 12 44 58 87
56 0 56 94 62 65 18 38 67 22
34 53 0 91 13 73 70 51 13 37
28 19 14 0 83 89 25 9 89 22
67 13 1 19 0 51 7 13 31 4
3 78 24 90 14 0 77 6 35 69
96 32 100 4 8 19 0 26 37 36
79 2 48 25 63 99 17 0 44 45
97 49 33 74 23 72 23 73 0 87
58 83 24 39 17 76 64 78 100 0

Минимальная стоимость: 127
Оптимальный путь: 0 -> 3 -> 7 -> 1 -> 9 -> 4 -> 2 -> 8 -> 6 -> 5 -> 0
```

Решение руками

Задача коммивояжера. Возьмем в качестве произвольного маршрута:

$$X_0 = (1,2); (2,3); (3,4); (4,5); (5,6); (6,7); (7,8); (8,9); (9,10); (10,1)$$

Тогда $F(X_0) = 34+56+91+83+51+77+26+44+87+58 = 607$ Для определения нижней границы множества воспользуемся **операцией редукции** или приведения матрицы по строкам, для чего необходимо в каждой строке матрицы D найти минимальный элемент.

$$d_i = \min_j d_{ij}$$

Затем вычитаем d_i из элементов рассматриваемой строки. В связи с этим во вновь полученной матрице в каждой строке будет как минимум один ноль. Такую же операцию редукции проводим по столбцам, для чего в каждом столбце находим минимальный элемент:

$$d_j = \min_i d_{ij}$$

После вычитания минимальных элементов получаем полностью редуцированную матрицу, где величины d_i и d_j называются **константами приведения**.

i j	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	M	22	56	6	51	53	0	32	46	72
2	38	M	38	76	44	32	0	20	49	1
3	21	40	M	78	0	45	57	38	0	21
4	19	10	5	M	74	65	16	0	80	10
5	66	12	0	18	M	35	6	12	30	0
6	0	75	21	87	11	M	74	3	32	63
7	92	28	96	0	4	0	M	22	33	29
8	77	0	46	23	61	82	15	M	42	40
9	74	26	10	51	0	34	0	50	M	61
10	41	66	7	22	0	44	47	61	83	M

Сумма констант приведения определяет нижнюю границу H:

$$H = \sum d_i + \sum d_j$$

$$H = 12+18+13+9+1+3+4+2+23+17+0+0+0+0+0+15+0+0+0+3 = 120$$

Шаг №1. Определяем ребро ветвления и разобьем все множество маршрутов относительно этого ребра на два подмножества (i,j) и (i^*,j^*). С этой целью для всех клеток матрицы с нулевыми элементами заменяем поочередно нули на M (бесконечность) и определяем для них сумму образовавшихся констант приведения, они приведены в скобках.

$$d(1,7) = 6+0 = 6;$$
$$d(2,7) = 1+0 = 1;$$
$$d(3,5) = 0+0 = 0;$$
$$d(3,9) = 0+30 = 30;$$
$$d(4,8) = 5+3 = 8;$$
$$d(5,3) = 0+5 = 5;$$
$$d(5,10) = 0+1 = 1;$$
$$d(6,1) = 3+19 = 22;$$
$$d(7,4) = 0+6 = 6;$$
$$d(7,6) = 0+32 = 32;$$
$$d(8,2) = 15+10 = 25;$$
$$d(9,5) = 0+0 = 0;$$
$$d(9,7) = 0+0 = 0;$$
$$d(10,5) = 7+0 = 7;$$

Наибольшая сумма констант приведения равна $(0 + 32) = 32$ для ребра $(7, 6)$, следовательно, множество разбивается на два подмножества $(7, 6)$ и $(7^*, 6^*)$. **Исключение ребра** $(7, 6)$ проводим путем замены элемента $d_{76} = 0$ на M, после чего осуществляем очередное приведение матрицы расстояний для образовавшегося подмножества $(7^*, 6^*)$, в результате получим редуцированную матрицу.

i j	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**di**
1	M	22	56	6	51	53	0	32	46	72	0
2	38	M	38	76	44	32	0	20	49	1	0
3	21	40	M	78	0	45	57	38	0	21	0
4	19	10	5	M	74	65	16	0	80	10	0
5	66	12	0	18	M	35	6	12	30	0	0
6	0	75	21	87	11	M	74	3	32	63	0
7	92	28	96	0	4	M	M	22	33	29	0
8	77	0	46	23	61	82	15	M	42	40	0
9	74	26	10	51	0	34	0	50	M	61	0
10	41	66	7	22	0	44	47	61	83	M	0
dj	0	0	0	0	0	32	0	0	0	0	32

Нижняя граница гамильтоновых циклов этого подмножества:

$$H(7^*, 6^*) = 120 + 32 = 152$$

Включение ребра $(7, 6)$ проводится путем исключения всех элементов 7-ой строки и 6-го столбца, в которой элемент d_{67} заменяем на M, для исключения образования негамильтонова цикла. В результате получим другую сокращенную матрицу (9 x 9), которая подлежит операции приведения. После операции приведения сокращенная матрица будет иметь вид:

i j	**1**	**2**	**3**	**4**	**5**	**7**	**8**	**9**	**10**	**di**
1	M	22	56	6	51	0	32	46	72	0
2	38	M	38	76	44	0	20	49	1	0
3	21	40	M	78	0	57	38	0	21	0
4	19	10	5	M	74	16	0	80	10	0
5	66	12	0	18	M	6	12	30	0	0
6	0	75	21	87	11	M	3	32	63	0
8	77	0	46	23	61	15	M	42	40	0
9	74	26	10	51	0	0	50	M	61	0
10	41	66	7	22	0	47	61	83	M	0
dj	0	0	0	6	0	0	0	0	0	6

Сумма констант приведения сокращенной матрицы:

$$\sum d_i + \sum d_j = 6$$

Нижняя граница подмножества $(7, 6)$ равна:

$$H(7, 6) = 120 + 6 = 126 \leq 152$$

Поскольку нижняя граница этого подмножества $(7, 6)$ меньше, чем подмножества $(7^*, 6^*)$, то ребро $(7, 6)$ включаем в маршрут с новой границей $H = 126$.

Шаг №2. Определяем ребро ветвления.

$$d(1,4) = 0 + 12 = 12;$$
$$d(1,7) = 0 + 0 = 0;$$
$$d(2,7) = 1 + 0 = 1;$$
$$d(3,5) = 0 + 0 = 0;$$
$$d(3,9) = 0 + 30 = 30;$$
$$d(4,8) = 5 + 3 = 8;$$
$$d(5,3) = 0 + 5 = 5;$$
$$d(5,10) = 0 + 1 = 1;$$
$$d(6,1) = 3 + 19 = 22;$$
$$d(8,2) = 15 + 10 = 25;$$
$$d(9,5) = 0 + 0 = 0;$$
$$d(9,7) = 0 + 0 = 0;$$
$$d(10,5) = 7 + 0 = 7;$$

$$\max\colon d(3,9) = 30.$$

Исключение ребра $(3,9)$: $d_{39} = M$.

i j	**1**	**2**	**3**	**4**	**5**	**7**	**8**	**9**	**10**	**di**
1	M	22	56	0	51	0	32	46	72	0
2	38	M	38	70	44	0	20	49	1	0
3	21	40	M	72	0	57	38	M	21	0
4	19	10	5	M	74	16	0	80	10	0
5	66	12	0	12	M	6	12	30	0	0
6	0	75	21	81	11	M	3	32	63	0
8	77	0	46	17	61	15	M	42	40	0
9	74	26	10	45	0	0	50	M	61	0
10	41	66	7	16	0	47	61	83	M	0
dj	0	0	0	0	0	0	0	30	0	30

$$H(3^*, 9^*) = 126 + 30 = 156$$

Включение ребра $(3,9)$: $d_{93} = M$.

i j	**1**	**2**	**3**	**4**	**5**	**7**	**8**	**10**	**di**
1	M	22	56	0	51	0	32	72	0
2	38	M	38	70	44	0	20	1	0
4	19	10	5	M	74	16	0	10	0
5	66	12	0	12	M	6	12	0	0
6	0	75	21	81	11	M	3	63	0
8	77	0	46	17	61	15	M	40	0
9	74	26	M	45	0	0	50	61	0
10	41	66	7	16	0	47	61	M	0
dj	0	0	0	0	0	0	0	0	0

$$\sum d_i + \sum d_j = 0$$

$$H(3,9) = 126 + 0 = 126 \leq 156$$

Ребро $(3,9)$ включаем в маршрут с новой границей $H = 126$. **Шаг №3. Определяем ребро ветвления**.

$$d(1,4) = 0 + 12 = 12;$$
$$d(1,7) = 0 + 0 = 0;$$
$$d(2,7) = 1 + 0 = 1;$$
$$d(4,8) = 5 + 3 = 8;$$
$$d(5,3) = 0 + 5 = 5;$$
$$d(5,10) = 0 + 1 = 1;$$
$$d(6,1) = 3 + 19 = 22;$$
$$d(8,2) = 15 + 10 = 25;$$
$$d(9,5) = 0 + 0 = 0;$$
$$d(9,7) = 0 + 0 = 0;$$
$$d(10,5) = 7 + 0 = 7;$$

$$\max\text{: } d(8,2) = 25.$$

Исключение ребра $(8,2)$: $d_{82} = M$.

i j	**1**	**2**	**3**	**4**	**5**	**7**	**8**	**10**	**di**
1	M	22	56	0	51	0	32	72	0
2	38	M	38	70	44	0	20	1	0
4	19	10	5	M	74	16	0	10	0
5	66	12	0	12	M	6	12	0	0
6	0	75	21	81	11	M	3	63	0
8	77	M	46	17	61	15	M	40	15
9	74	26	M	45	0	0	50	61	0
10	41	66	7	16	0	47	61	M	0
dj	0	10	0	0	0	0	0	0	25

$$H(8^*, 2^*) = 126 + 25 = 151$$

Включение ребра $(8,2)$: $d_{28} = M$.

i j	**1**	**3**	**4**	**5**	**7**	**8**	**10**	**di**
1	M	56	0	51	0	32	72	0
2	38	38	70	44	0	M	1	0
4	19	5	M	74	16	0	10	0
5	66	0	12	M	6	12	0	0
6	0	21	81	11	M	3	63	0
9	74	M	45	0	0	50	61	0
10	41	7	16	0	47	61	M	0
dj	0	0	0	0	0	0	0	0

$$\sum d_i + \sum d_j = 0$$

$$H(8,2) = 126 + 0 = 126 \leq 151$$

Ребро $(8,2)$ включаем в маршрут с новой границей $H = 126$.

Шаг №4. Определяем ребро ветвления.

$$d(1,4) = 0 + 12 = 12;$$
$$d(1,7) = 0 + 0 = 0;$$
$$d(2,7) = 1 + 0 = 1;$$
$$d(4,8) = 5 + 3 = 8;$$
$$d(5,3) = 0 + 5 = 5;$$
$$d(5,10) = 0 + 1 = 1;$$
$$d(6,1) = 3 + 19 = 22;$$
$$d(9,5) = 0 + 0 = 0;$$
$$d(9,7) = 0 + 0 = 0;$$
$$d(10,5) = 7 + 0 = 7;$$

$$\max\colon d(6,1) = 22.$$

Исключение ребра $(6,1)$: $d_{61} = M$.

i j	**1**	**3**	**4**	**5**	**7**	**8**	**10**	**di**
1	M	56	0	51	0	32	72	0
2	38	38	70	44	0	M	1	0
4	19	5	M	74	16	0	10	0
5	66	0	12	M	6	12	0	0
6	M	21	81	11	M	3	63	3
9	74	M	45	0	0	50	61	0
10	41	7	16	0	47	61	M	0
dj	19	0	0	0	0	0	0	22

$$H(6^*, 1^*) = 126 + 22 = 148$$

Включение ребра $(6,1)$: $d_{16} = M$.

i j	**3**	**4**	**5**	**7**	**8**	**10**	**di**
1	56	0	51	0	32	72	0
2	38	70	44	0	M	1	0
4	5	M	74	16	0	10	0
5	0	12	M	6	12	0	0
9	M	45	0	0	50	61	0
10	7	16	0	47	61	M	0
dj	0	0	0	0	0	0	0

$$\sum d_i + \sum d_j = 0$$
$$H(6,1) = 126 + 0 = 126 \leq 148$$

Запрещаем переходы: $(1,7)$, Ребро $(6,1)$ включаем в маршрут с новой границей $H = 126$.

Шаг №5. Определяем ребро ветвления.

$$d(1,4) = 32 + 12 = 44;$$
$$d(2,7) = 1 + 0 = 1;$$
$$d(4,8) = 5 + 12 = 17;$$
$$d(5,3) = 0 + 5 = 5;$$
$$d(5,10) = 0 + 1 = 1;$$
$$d(9,5) = 0 + 0 = 0;$$
$$d(9,7) = 0 + 0 = 0;$$
$$d(10,5) = 7 + 0 = 7;$$

$$\max\colon d(1,4) = 44.$$

Исключение ребра $(1,4)$: $d_{14} = M$.

i j	3	4	5	7	8	10	di
1	56	M	51	M	32	72	32
2	38	70	44	0	M	1	0
4	5	M	74	16	0	10	0
5	0	12	M	6	12	0	0
9	M	45	0	0	50	61	0
10	7	16	0	47	61	M	0
dj	0	12	0	0	0	0	44

$$H(1^*, 4^*) = 126 + 44 = 170$$

Включение ребра $(1, 4)$: $d_{41} = M$.

i j	3	5	7	8	10	di
2	38	44	0	M	1	0
4	5	74	16	0	10	0
5	0	M	6	12	0	0
9	M	0	0	50	61	0
10	7	0	47	61	M	0
dj	0	0	0	0	0	0

$$\sum d_i + \sum d_j = 0$$

$$H(1, 4) = 126 + 0 = 126 \leq 170$$

Запрещаем переходы: $(4, 7)$, $(4, 6)$, Ребро $(1, 4)$ включаем в маршрут с новой границей $H = 126$.

Шаг №6. Определяем ребро ветвления.

$$d(2, 7) = 1 + 0 = 1;$$
$$d(4, 8) = 5 + 12 = 17;$$
$$d(5, 3) = 0 + 5 = 5;$$
$$d(5, 10) = 0 + 1 = 1;$$
$$d(9, 5) = 0 + 0 = 0;$$
$$d(9, 7) = 0 + 0 = 0;$$
$$d(10, 5) = 7 + 0 = 7;$$

$$\max\text{:}\ d(4, 8) = 17.$$

Исключение ребра $(4, 8)$: $d_{48} = M$.

i j	3	5	7	8	10	di
2	38	44	0	M	1	0
4	5	74	M	M	10	5
5	0	M	6	12	0	0
9	M	0	0	50	61	0
10	7	0	47	61	M	0
dj	0	0	0	12	0	17

$$H(4^*, 8^*) = 126 + 17 = 143$$

Включение ребра $(4, 8)$: $d_{84} = M$.

i j	3	5	7	10	di
2	38	44	0	1	0
5	0	M	6	0	0
9	M	0	0	61	0
10	7	0	47	M	0
dj	0	0	0	0	0

$$\sum d_i + \sum d_j = 0$$

$$H(4,8) = 126 + 0 = 126 \leq 143$$

Запрещаем переходы: $(2,7)$, $(2,6)$, $(2,1)$, $(2,4)$, Ребро $(4,8)$ включаем в маршрут с новой границей $H = 126$.

Шаг №7. Определяем ребро ветвления.

i j	**3**	**5**	**7**	**10**	**di**
2	38	44	M	1	0
5	**0(7)**	M	6	0(1)	0
9	M	0(0)	0(6)	61	0
10	7	0(7)	47	M	7
dj	7	0	6	1	0

$$d(5,3) = 0 + 7 = 7;$$
$$d(5,10) = 0 + 1 = 1;$$
$$d(9,5) = 0 + 0 = 0;$$
$$d(9,7) = 0 + 6 = 6;$$
$$d(10,5) = 7 + 0 = 7;$$

$$\max\text{: } d(5,3) = 7.$$

Исключение ребра $(5,3)$: $d_{53} = M$.

i j	**3**	**5**	**7**	**10**	**di**
2	38	44	M	1	1
5	M	M	6	0	0
9	M	0	0	61	0
10	7	0	47	M	0
dj	7	0	0	0	8

$$H(5^*,3^*) = 126 + 8 = 134$$

Включение ребра $(5,3)$: $d_{35} = M$.

i j	**5**	**7**	**10**	**di**
2	44	M	1	1
9	0	0	61	0
10	0	47	M	0
dj	0	0	1	2

$$\sum d_i + \sum d_j = 2$$

$$H(5,3) = 126 + 2 = 128 \leq 134$$

Запрещаем переходы: $(2,7)$, $(2,6)$, $(2,1)$, $(2,4)$, $(9,5)$, Ребро $(5,3)$ включаем в маршрут с новой границей $H = 128$.

Шаг №8. Определяем ребро ветвления.

i j	**5**	**7**	**10**	**di**
2	43	M	0(104)	43
9	M	**0(108)**	61	61
10	0(90)	47	M	47
dj	43	47	61	0

$$d(2,10) = 43 + 61 = 104;$$
$$d(9,7) = 61 + 47 = 108;$$
$$d(10,5) = 47 + 43 = 90;$$

$$\max\text{: } d(9,7) = 108.$$

Исключение ребра $(9,7)$: $d_{97} = M$.

i j	5	7	10	di
2	43	M	0	0
9	M	M	61	61
10	0	47	M	0
dj	0	47	0	108

$$H(9^*, 7^*) = 128 + 108 = 236$$

Включение ребра $(9,7)$: $d_{79} = M$.

i j	5	10	di
2	43	0	0
10	0	M	0
dj	0	0	0

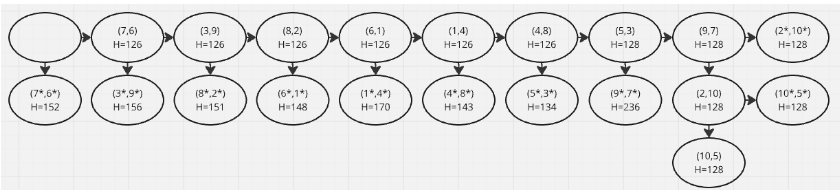
$$\sum d_i + \sum d_j = 0$$

$$H(9,7) = 128 + 0 = 128 \leq 236$$

Ребро $(9,7)$ включаем в маршрут с новой границей $H = 128$. В соответствии с этой матрицей включаем в гамильтонов маршрут ребра $(2,10)$ и $(10,5)$. В результате по дереву ветвлений гамильтонов цикл образуют ребра:

$$(7,6), (6,1), (1,4), (4,8), (8,2), (2,10), (10,5), (5,3), (3,9), (9,7),$$

Длина маршрута равна $F(M_k) = 127$.



Работа №8

Решение задачи о назначениях с помощью Венгерского алгоритма

Код решения:

```
from random import randint

import numpy as np
from scipy.optimize import linear_sum_assignment


def hungarian_algorithm(cost_matrix):
    cost_matrix = np.array(cost_matrix)

    # Check if the matrix is rectangular and non-empty
    if len(cost_matrix.shape) != 2 or cost_matrix.shape[0] == 0 or cost_matrix.shape[1] == 0:
        raise ValueError("Cost matrix must be a non-empty 2D array.")

    row_indices, col_indices = linear_sum_assignment(cost_matrix)

    return row_indices.tolist(), col_indices.tolist()


def main():
    try:
        n = int(input())
        should_be_generated = input("Write something to generate table")
        if should_be_generated:
            cost_matrix = [[randint(1, 9) for j in range(n)] for _ in range(n)]
            for i in cost_matrix:
                print(*i)
        else:
            cost_matrix = [list(map(int, input().split())) for _ in range(n)]
        row_ind, col_ind = hungarian_algorithm(cost_matrix)
        print("Optimal assignment:")
        optimal_assignment = [[0 for j in range(n)] for i in range(n)]
        for r, c in zip(row_ind, col_ind):
            optimal_assignment[r][c] = 1
        for i in optimal_assignment:
            print(*i)

        total_cost = sum(cost_matrix[r][c] for r, c in zip(row_ind, col_ind))
        print(f"Total cost: {total_cost}")
    except Exception as e:
        print(f"Error: {e}")


if __name__ == "__main__":
    main()

    Ввод-вывод:
    10
    1 6 5 5 8 2 8 4 6 8
    9 9 1 6 3 2 4 8 9 9
    2 5 2 6 8 8 2 3 9 7
    8 8 8 6 7 9 5 2 2 2
    5 2 7 5 7 4 7 9 4 2
    1 3 7 7 2 8 7 1 4 7
```

```
5 6 9 2 3 6 8 5 8 3
8 9 6 7 5 9 2 3 7 3
2 2 2 9 8 8 2 1 5 1
2 6 8 4 8 6 7 9 5 7
Optimal assignment:
0 0 0 0 0 1 0 0 0 0
0 0 1 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 1 0
0 1 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0
0 0 0 1 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 1
1 0 0 0 0 0 0 0 0 0
Total cost: 19
```

Исходная матрица

Исходная матрица имеет вид:

$$\begin{matrix}
1 & 6 & 5 & 5 & 8 & 2 & 8 & 4 & 6 & 8 \\
9 & 9 & 1 & 6 & 3 & 2 & 4 & 8 & 9 & 9 \\
2 & 5 & 2 & 6 & 8 & 8 & 2 & 3 & 9 & 7 \\
8 & 8 & 8 & 6 & 7 & 9 & 5 & 2 & 2 & 2 \\
5 & 2 & 7 & 5 & 7 & 4 & 7 & 9 & 4 & 2 \\
1 & 3 & 7 & 7 & 2 & 8 & 7 & 1 & 4 & 7 \\
5 & 6 & 9 & 2 & 3 & 6 & 8 & 5 & 8 & 3 \\
8 & 9 & 6 & 7 & 5 & 9 & 2 & 3 & 7 & 3 \\
2 & 2 & 2 & 9 & 8 & 8 & 2 & 1 & 5 & 1 \\
2 & 6 & 8 & 4 & 8 & 6 & 7 & 9 & 5 & 7
\end{matrix}$$

Шаг №1

1. Проводим редукцию матрицы по строкам

В связи с этим во вновь полученной матрице в каждой строке будет как минимум один ноль.

$$\begin{matrix}
0 & 5 & 4 & 4 & 7 & 1 & 7 & 3 & 5 & 7 & \mathbf{1} \\
8 & 8 & 0 & 5 & 2 & 1 & 3 & 7 & 8 & 8 & \mathbf{1} \\
0 & 3 & 0 & 4 & 6 & 6 & 0 & 1 & 7 & 5 & \mathbf{2} \\
6 & 6 & 6 & 4 & 5 & 7 & 3 & 0 & 0 & 0 & \mathbf{2} \\
3 & 0 & 5 & 3 & 5 & 2 & 5 & 7 & 2 & 0 & \mathbf{2} \\
0 & 2 & 6 & 6 & 1 & 7 & 6 & 0 & 3 & 6 & \mathbf{1} \\
3 & 4 & 7 & 0 & 1 & 4 & 6 & 3 & 6 & 1 & \mathbf{2} \\
6 & 7 & 4 & 5 & 3 & 7 & 0 & 1 & 5 & 1 & \mathbf{2} \\
1 & 1 & 1 & 8 & 7 & 7 & 1 & 0 & 4 & 0 & \mathbf{1} \\
0 & 4 & 6 & 2 & 6 & 4 & 5 & 7 & 3 & 5 & \mathbf{2}
\end{matrix}$$

Затем такую же операцию редукции проводим по столбцам, для чего в каждом столбце находим минимальный элемент.

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

После вычитания минимальных элементов получаем полностью редуцированную матрицу.

2. Методом проб и ошибок проводим поиск допустимого решения

Фиксируем нулевое значение в клетке (1, 6). Другие нули в строке 1 и столбце 6 вычеркиваем. Для данной клетки вычеркиваем нули в клетках (2; 6), (1; 1).

[0]	5	4	4	6	**[0]**	7	3	5	7
8	8	**[0]**	5	1	[0]	3	7	8	8
[0]	3	[0]	4	5	5	**[0]**	1	7	5
6	6	6	4	4	6	3	[0]	[0]	**[0]**
3	**[0]**	5	3	4	1	5	7	2	[0]
[0]	2	6	6	[0]	6	6	**[0]**	3	6
3	4	7	[0]	**[0]**	3	6	3	6	1
6	7	4	5	2	6	[0]	1	5	1
1	1	1	8	6	6	1	[0]	4	[0]
0	4	6	2	5	3	5	7	3	5

Поскольку расположение нулевых элементов в матрице не позволяет образовать систему из 10 независимых нулей (в матрице их только 7), то решение недопустимое.

3. Проводим модификацию матрицы

Вычеркиваем строки и столбцы с возможно большим количеством нулевых элементов: столбец 1, строку 4, строку 2, столбец 5, строку 3, столбец 8, строку 5, столбец 4, строку 1, столбец 7, строку 9.

Получаем сокращенную матрицу (элементы выделены):

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

Минимальный элемент сокращенной матрицы (min(2, 6, 6, 3, 6, 4, 7, 3, 6, 1, 7, 4, 6, 5, 1, 4, 6, 3, 3, 5) = 1) вычитаем из всех ее элементов:

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

Затем складываем минимальный элемент с элементами, расположенными на пересечениях вычеркнутых строк и столбцов:

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

Шаг №2

1. Проводим редукцию матрицы по строкам

В связи с этим во вновь полученной матрице в каждой строке будет как минимум один ноль.

1	5	4	5	7	**[0]**	8	4	5	7
9	8	**[0]**	6	2	[0]	4	8	8	8
1	3	[0]	5	6	5	1	2	7	5
7	6	6	5	5	6	4	1	0	0
4	0	5	4	5	1	6	8	2	0
0	1	5	6	0	5	6	0	2	5
3	3	6	0	0	2	6	3	5	0
6	6	3	5	2	5	0	1	4	0
2	1	1	9	7	6	2	1	4	0
0	3	5	2	5	2	5	7	3	4

Поскольку расположение нулевых элементов в матрице не позволяет образовать систему из 10 независимых нулей (в матрице их только 2), то решение недопустимое.

3. Проводим модификацию матрицы

Вычеркиваем строки и столбцы с возможно большим количеством нулевых элементов: столбец 10, строку 6, строку 2, строку 7, столбец 1, столбец 2, строку 1, столбец 3, строку 4, столбец 7.

Получаем сокращенную матрицу (элементы выделены):

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

Минимальный элемент сокращенной матрицы (min(5, 6, 5, 2, 7, 4, 5, 1, 8, 2, 5, 2, 5, 1, 4, 9, 7, 6, 1, 4, 2, 5, 2, 7, 2) = 1) вычитаем из всех ее элементов:

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

Затем складываем минимальный элемент с элементами, расположенными на пересечениях вычеркнутых строк и столбцов:

2	**6**	**5**	5	7	0	**9**	4	5	**8**
10	**9**	**1**	6	2	0	**5**	8	8	**9**
1	3	0	4	**6**	**5**	1	**2**	6	**6**
8	**7**	**7**	5	5	6	**5**	1	0	**1**
4	0	5	3	**5**	**1**	6	**8**	1	**1**
1	**2**	**6**	6	0	**5**	**7**	0	2	**6**
4	**4**	**7**	0	**1**	**3**	**7**	3	5	**2**
7	**7**	**4**	5	**2**	**5**	0	**1**	4	**1**
2	**1**	**1**	**9**	**7**	**6**	2	**1**	4	**0**
0	**3**	**5**	2	5	**2**	5	**7**	3	**4**

Шаг №3

1. Проводим редукцию матрицы по строкам

В связи с этим во вновь полученной матрице в каждой строке будет как минимум один ноль.

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

Поскольку расположение нулевых элементов в матрице не позволяет образовать систему из 10 независимых нулей (в матрице их только 1), то решение недопустимое.

3. Проводим модификацию матрицы

Вычеркиваем строки и столбцы с возможно большим количеством нулевых элементов: строку 5, столбец 8, строку 7, столбец 6, строку 8, столбец 1, строку 3, столбец 5, строку 4, столбец 10.

Получаем сокращенную матрицу (элементы выделены):

2	**6**	**5**	**5**	7	0	**9**	4	**5**	8
10	**9**	**1**	**6**	2	0	**5**	8	**8**	9
1	3	0	4	5	4	1	1	6	5
8	7	7	5	5	6	5	1	0	1
4	0	5	3	4	1	6	8	1	0
1	**2**	**6**	**6**	0	**5**	**7**	0	**2**	6
4	4	7	0	0	2	7	3	5	1
7	7	4	**5**	**2**	**5**	0	**1**	4	1
2	**1**	**1**	**9**	6	**6**	2	**1**	4	0
0	**3**	**5**	**2**	5	**2**	5	**7**	3	4

Минимальный элемент сокращенной матрицы (min(6, 5, 5, 9, 5, 9, 1, 6, 5, 8, 2, 6, 6, 7, 2, 1, 1, 8, 2, 3, 3, 5, 1, 5, 1) = 1) вычитаем из всех ее элементов:

2	**5**	**4**	**4**	7	0	**8**	4	**4**	8
10	**8**	**0**	**5**	2	0	**4**	8	**7**	9
1	3	0	4	5	4	1	1	6	5
8	7	7	5	5	6	5	1	0	1
4	0	5	3	4	1	6	8	1	0
1	**1**	**5**	**5**	0	**5**	**6**	0	**1**	6
4	4	7	0	0	2	7	3	5	1
7	7	4	**4**	**1**	**4**	0	**0**	**3**	1
2	**0**	**0**	**8**	6	**5**	2	**0**	4	0
0	**2**	**4**	**1**	4	**1**	5	**6**	**0**	4

Затем складываем минимальный элемент с элементами, расположенными на пересечениях вычеркнутых строк и столбцов:

2	5	4	4	7	0	8	4	4	8
10	8	0	5	2	0	4	8	7	9
2	3	0	4	**6**	**5**	1	**2**	6	**6**
9	7	7	5	**6**	**7**	5	**2**	0	**2**
5	0	5	3	**5**	**1**	6	**8**	1	**1**
1	1	5	5	0	5	6	0	1	6
5	4	7	0	**1**	**3**	7	**4**	5	**2**
8	7	4	5	**2**	**5**	0	**1**	4	**1**
2	0	0	8	6	5	1	0	4	0
0	2	4	1	4	1	5	6	1	4

Шаг №4

1. Проводим редукцию матрицы по строкам. В связи с этим во вновь полученной матрице в каждой строке будет как минимум один ноль. Затем такую же операцию редукции проводим по столбцам, для чего в каждом столбце находим минимальный элемент. После вычитания минимальных элементов получаем полностью редуцированную матрицу.

2. Методом проб и ошибок проводим поиск допустимого решения, для которого все назначения имеют нулевую стоимость. Фиксируем нулевое значение в клетке (1, 6). Другие нули в строке 1 и столбце 6 вычеркиваем. Для данной клетки вычеркиваем нули в клетках (2; 6). Фиксируем нулевое значение в клетке (2, 3). Другие нули в строке 2 и столбце 3 вычеркиваем. Для данной клетки вычеркиваем нули в клетках (3; 3), (9; 3).

В итоге получаем следующую матрицу:

2	5	4	4	7	**[0]**	8	4	4	8
10	8	**[0]**	5	2	[−0−]	4	8	7	9
2	3	[−0−]	4	6	5	1	2	6	6
9	7	7	5	6	7	5	2	0	2
5	0	5	3	5	1	6	8	1	1
1	1	5	5	0	5	6	0	1	6
5	4	7	0	1	3	7	4	5	2
7	6	3	4	2	5	0	1	3	1
2	0	[−0−]	7	6	5	1	0	2	0
0	2	4	0	4	1	4	6	0	4

Поскольку расположение нулевых элементов в матрице не позволяет образовать систему из 10 независимых нулей (в матрице их только 2), то **решение недопустимое**.

3. Проводим модификацию матрицы. Вычеркиваем строки и столбцы с возможно большим количеством нулевых элементов: строку 9, строку 10, столбец 3, столбец 6, строку 6, столбец 2, строку 4, столбец 4, строку 8.

Получаем сокращенную матрицу (элементы выделены):

2	5	4	4	**7**	0	**8**	**4**	**4**	**8**
10	8	0	5	**2**	0	**4**	**8**	**7**	**9**
2	3	0	4	**6**	5	**1**	**2**	**6**	**6**
9	7	7	5	6	7	5	2	0	2
5	0	5	3	**5**	1	**6**	**8**	**1**	**1**
1	1	5	5	0	5	6	0	1	6
5	4	7	0	**1**	3	**7**	**4**	**5**	**2**
7	6	3	4	2	5	0	1	3	1
2	0	0	7	6	5	1	0	2	0
0	2	4	0	4	1	4	6	0	4

Минимальный элемент сокращенной матрицы ($\min(2, 7, 8, 4, 4, 8, 10, 2, 4, 8, 7, 9, 2, 6, 1, 2, 6, 6, 5, 5, 6, 8, 1, 1, 5, 1, 7, 4, 5$
1) вычитаем из всех ее элементов:

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

Затем складываем минимальный элемент с элементами, расположенными на пересечениях вычеркнутых строк и столбцов:

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

Шаг №5

1. Проводим редукцию матрицы по строкам. В связи с этим во вновь полученной матрице в каждой строке будет как минимум один ноль. Затем такую же операцию редукции проводим по столбцам, для чего в каждом столбце находим минимальный элемент. После вычитания минимальных элементов получаем полностью редуцированную матрицу.

2. Методом проб и ошибок проводим поиск допустимого решения, для которого все назначения имеют нулевую стоимость. Фиксируем нулевое значение в клетке (1, 6). Другие нули в строке 1 и столбце 6 вычеркиваем. Для данной клетки вычеркиваем нули в клетках (2; 6). Фиксируем нулевое значение в клетке (2, 3). Другие нули в строке 2 и столбце 3 вычеркиваем. Для данной клетки вычеркиваем нули в клетках (3; 3).

В итоге получаем следующую матрицу:

1	5	4	4	6	**[0]**	7	3	3	7
9	8	**[0]**	5	1	[−0−]	3	7	6	8
1	3	[−0−]	4	5	5	**[0]**	1	5	5
9	8	8	6	6	8	5	2	**[0]**	2
4	[−0−]	5	3	4	1	5	7	[−0−]	**[0]**
1	2	6	6	[−0−]	6	6	**[0]**	1	6
4	4	7	[−0−]	**[0]**	3	6	3	4	1
7	7	4	5	2	6	[−0−]	1	3	1
2	1	1	8	6	6	1	[−0−]	2	[−0−]
0	3	5	1	4	2	4	6	[−0−]	4

Поскольку расположение нулевых элементов в матрице не позволяет образовать систему из 10 независимых нулей (в матрице их только 7), то **решение недопустимое**.

3. Проводим модификацию матрицы. Вычеркиваем строки и столбцы с возможно большим количеством нулевых элементов: строку 5, столбец 3, строку 6, столбец 6, строку 7, столбец 7, строку 9, столбец 9, строку 10.

Получаем сокращенную матрицу (элементы выделены):

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

Минимальный элемент сокращенной матрицы ($\min(1, 5, 4, 6, 3, 7, 9, 8, 5, 1, 7, 8, 1, 3, 4, 5, 1, 5, 9, 8, 6, 6, 2, 2, 7, 7, 5, 2, 1,$ 1) вычитаем из всех ее элементов:

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

Затем складываем минимальный элемент с элементами, расположенными на пересечениях вычеркнутых строк и столбцов:

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

Шаг №6

1. Проводим редукцию матрицы по строкам. В связи с этим во вновь полученной матрице в каждой строке будет как минимум один ноль. Затем такую же операцию редукции проводим по столбцам, для чего в каждом столбце находим минимальный элемент. После вычитания минимальных элементов получаем полностью редуцированную матрицу.

2. Методом проб и ошибок проводим поиск допустимого решения, для которого все назначения имеют нулевую стоимость. Фиксируем нулевое значение в клетке (1, 6). Другие нули в строке 1 и столбце 6 вычеркиваем. Для данной клетки вычеркиваем нули в клетках (2; 6), (1; 1).

В итоге получаем следующую матрицу:

[−0−]	4	4	3	5	**[0]**	7	2	3	6
8	7	**[0]**	4	[−0−]	[−0−]	3	6	6	7
[−0−]	2	[−0−]	3	4	5	**[0]**	[−0−]	5	4
8	7	8	5	5	8	5	1	**[0]**	1
4	**[0]**	6	3	4	2	6	7	1	[−0−]
1	2	7	6	**[0]**	7	7	[−0−]	2	6
4	4	8	**[0]**	[−0−]	4	7	3	5	1
6	6	4	4	1	6	**[0]**	[−0−]	3	[−0−]
2	1	2	8	6	7	2	[−0−]	3	**[0]**
[0]	3	6	1	4	3	5	6	1	4

Количество найденных нулей равно $k = 10$. В результате получаем эквивалентную матрицу C_e:

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

4. Методом проб и ошибок определяем матрицу назначения X, которая позволяет по аналогично расположенным элементам исходной матрицы (в квадратах) вычислить минимальную стоимость назначения.

[−0−]	4	4	3	5	**[0]**	7	2	3	6
8	7	**[0]**	4	[−0−]	[−0−]	3	6	6	7
[−0−]	2	[−0−]	3	4	5	[−0−]	**[0]**	5	4
8	7	8	5	5	8	5	1	**[0]**	1
4	**[0]**	6	3	4	2	6	7	1	[−0−]
1	2	7	6	**[0]**	7	7	[−0−]	2	6
4	4	8	**[0]**	[−0−]	4	7	3	5	1
6	6	4	4	1	6	**[0]**	[−0−]	3	[−0−]
2	1	2	8	6	7	2	[−0−]	3	**[0]**
[0]	3	6	1	4	3	5	6	1	4

Ответ:

$$C_{\min} = 2 + 2 + 2 + 2 + 2 + 1 + 2 + 2 + 1 + 3 = 19$$

Путь: (4;9), (5;2), (7;4), (1;6), (10;1), (2;3), (6;5), (8;7), (9;10), (3;8) или цикл: (4;9), (9;10), (10;1), (1;6), (6;5), (5;2), (2;3), (3;8), (8;7), (7;4)

Работа №9

Решение задачи распределния ресурсов

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

$$\phi_1(x) = \max\{f_1(x_1)\colon x_1 \in \overline{0,x}\}$$

$$\phi_1(0) = 0, x_1^0 = 0$$

$$\phi_1(1) = 3, x_1^0 = 1$$

$$\phi_1(2) = 4, x_1^0 = 2$$

$$\phi_1(3) = 8, x_1^0 = 3$$

$$\phi_1(4) = 8, x_1^0 = 3$$

$$\phi_1(5) = 9, x_1^0 = 5$$

$$\phi_1(6) = 9, x_1^0 = 5$$

$$\phi_2(x) = \max\{f_2(x_2) + \phi_1(x - x_2)\colon x_2 \in \overline{0,x}\}$$

Текущее множество: $(3, 4)$ $\quad \phi_2(1) = 4, x_2^0 = 1$

Текущее множество: $(4, 7, 3)$ $\quad \phi_2(2) = 7, x_2^0 = 1$

Текущее множество: $(8, 8, 6, 5)$ $\quad \phi_2(3) = 8, x_2^0 = 0$

Текущее множество: $(8, 12, 7, 8, 7)$ $\quad \phi_2(4) = 12, x_2^0 = 1$

Текущее множество: $(9, 12, 11, 9, 10, 3)$ $\quad \phi_2(5) = 12, x_2^0 = 1$

Текущее множество: $(9, 13, 11, 13, 11, 6, 7)$ $\quad \phi_2(6) = 13, x_2^0 = 1$

$$\phi_3(x) = \max\{f_3(x_3) + \phi_2(x - x_3)\colon x_3 \in \overline{0,x}\}$$

Текущее множество: $(4, 7)$ $\quad \phi_3(1) = 7, x_3^0 = 1$

Текущее множество: $(7, 11, 5)$ $\quad \phi_3(2) = 11, x_3^0 = 1$

Текущее множество: $(8, 14, 9, 1)$ $\quad \phi_3(3) = 14, x_3^0 = 1$

Текущее множество: $(12, 15, 12, 5, 2)$ $\quad \phi_3(4) = 15, x_3^0 = 1$

Текущее множество: $(12, 19, 13, 8, 6, 3)$ $\quad \phi_3(5) = 19, x_3^0 = 1$

Текущее множество: $(13, 19, 17, 9, 9, 7, 5)$ $\quad \phi_3(6) = 19, x_3^0 = 1$

$$\phi_4(x) = \max\{f_4(x_4) + \phi_3(x - x_4)\colon x_4 \in \overline{0,x}\}$$

Текущее множество: $(7, 7)$ $\quad \phi_4(1) = 7, x_4^0 = 0$

Текущее множество: $(11, 14, 8)$ $\quad \phi_4(2) = 14, x_4^0 = 1$

Текущее множество: $(14, 18, 15, 9)$ $\quad \phi_4(3) = 18, x_4^0 = 1$

Текущее множество: $(15, 21, 19, 16, 4)$ $\quad \phi_4(4) = 21, x_4^0 = 1$

Текущее множество: $(19, 22, 22, 20, 11, 6)$ $\quad \phi_4(5) = 22, x_4^0 = 1$

Текущее множество: $(19, 26, 23, 23, 15, 13, 9)$ $\quad \phi_4(6) = 26, x_4^0 = 1$

$$\phi_5(x) = \max\{f_5(x_5) + \phi_4(x - x_5) \colon x_5 \in \overline{0, x}\}$$

Текущее множество: $(7, 2) \quad \phi_5(1) = 7, x_5^0 = 0$

Текущее множество: $(14, 9, 1) \quad \phi_5(2) = 14, x_5^0 = 0$

Текущее множество: $(18, 16, 8, 2) \quad \phi_5(3) = 18, x_5^0 = 0$

Текущее множество: $(21, 20, 15, 9, 9) \quad \phi_5(4) = 21, x_5^0 = 0$

Текущее множество: $(22, 23, 19, 16, 16, 2) \quad \phi_5(5) = 23, x_5^0 = 1$

Текущее множество: $(26, 24, 22, 20, 23, 9, 1) \quad \phi_5(6) = 26, x_5^0 = 0$

$$\phi_6(x) = \max\{f_6(x_6) + \phi_5(x - x_6) \colon x_6 \in \overline{0, x}\}$$

Текущее множество: $(7, 3) \quad \phi_6(1) = 7, x_6^0 = 0$

Текущее множество: $(14, 10, 2) \quad \phi_6(2) = 14, x_6^0 = 0$

Текущее множество: $(18, 17, 9, 5) \quad \phi_6(3) = 18, x_6^0 = 0$

Текущее множество: $(21, 21, 16, 12, 6) \quad \phi_6(4) = 21, x_6^0 = 0$

Текущее множество: $(23, 24, 20, 19, 13, 9) \quad \phi_6(5) = 24, x_6^0 = 1$

Текущее множество: $(26, 26, 23, 23, 20, 16, 1) \quad \phi_6(6) = 26, x_6^0 = 0$

$$6- > 0 \quad S = 6 \quad 5- > 0 \quad S = 6 \quad 4- > 1 \quad S = 5 \quad 3- > 1 \quad S = 4 \quad 2- > 1 \quad S = 3 \quad 1- > 3 \quad S = 0$$

$$x^o = (3, 1, 1, 1, 0, 0)$$

$$8 + 4 + 7 + 7 + 0 + 0 = 26$$

Программа для решения

```
import random

def generate_random_matrix(n):
    matrix = [[random.randint(1, 9) for j in range(n)] for i in range(n)]
    return matrix

def print_markdown_matrix(matrix):
    m = len(matrix[0])
    print("| " * m + "|")
    print("| --- " * m + "|")
    for i in range(len(matrix)):
        print(f"|", end='')
        for j in range(len(matrix[i])):
            print(matrix[i][j], end='|')
        print()
    print()

def input_matrix(n):
    matrix = []
    for i in range(n):
        row = list(map(int, input().split()))
        matrix.append(row)
    return matrix

def main():
    generate = input()
    N = int(input())
```

```
    n = N + 1
    if generate:
        matrix = generate_random_matrix(n)
    else:
        matrix = input_matrix(n)
    for i in range(n):
        matrix[i][0] = i
    for j in range(n):
        matrix[0][j] = 0
    print_markdown_matrix(matrix)
    phi = [[0 for j in range(n)] for i in range(n)]
    x_ = [[0] * n for _ in range(n)]
    print(r"$\phi_1(x)=\max\{f_1(x_1) \colon x_1 \in \overline{0,x}\}$")
    for x in range(n):
        set_ = [matrix[i][1] for i in range(x + 1)]
        for item_number in range(len(set_)):
            if set_[item_number] > set_[x_[1][x]]:
                x_[1][x] = item_number
        phi[1][x] = max(phi[1][x - 1], matrix[x][1])
        print(fr"$\phi_1({x})={phi[1][x]}, x_1^0={x_[1][x]}$")
    print()
    for number in range(2, n):
        print(rf"$\phi_{number}(x)=\max" + r"\{" + fr"f_{number}(x_{number}) + \phi_{number - 1}(x -
        for x in range(1, n):
            set_ = [matrix[i][number] + phi[number - 1][x - i] for i in range(x + 1)]
            for item_number in range(len(set_)):
                if set_[item_number] > set_[x_[number][x]]:
                    x_[number][x] = item_number
            phi[number][x] = max(set_)
            print(f"Current set: {tuple(set_)}")
            print(fr"$\phi_{number}({x})={phi[number][x]}, x_{number}^0={x_[number][x]}$")
        print()
    S = N
    i = N
    x_o = [0] * n
    while S > 0:
        x_o[i] = x_[i][S]
        print(f"{i}->{x_[i][S]}")
        S -= x_[i][S]
        print(f"$S={S}$")
        i -= 1
    print(f"$x^o=({", ".join(map(str, x_o[1:]))})$")
    print("$" + "+".join(str(matrix[x_o[i]][i]) for i in range(1, n)) + f" = {phi[N][N]}$")


if __name__ == "__main__":
    main()
```

Работа №10

Задача о рюкзаке

Имеется рюкзак грузоподъемностью W. $weight_i$ – вес одного предмета i-ого типа, $cost_i$ – стоимость (ценность) одного предмета i-ого типа, x_i – число предметов i-ого типа, которые будут загружаться на транспортировочное средство. Требуется заполнить его грузом, состоящим из предметов N различных типов таким образом, чтобы стоимость (ценность) всего груза была максимальной.

$$W(x) = \sum_{i=1}^{N} x_i \cdot cost_i \to max$$

$$\sum_{i=1}^{N} x_i \cdot weight_i \leq W, \quad x_i \in \{0\} \cup \mathbb{N}$$

Решение задачи разбивается на N этапов. На каждом i-ом этапе определяется максимальная стоимость груза, состоящего из предметов типа $k = \overline{1, i}$

Рекуррентное уравнение Беллмана для задачи о рюкзаке

$W_i(weight)$ - максимальная стоимость груза, состоящего из предметов типа $k = \overline{1, i}$ с общим весом не более $weight$.

$$\forall\ weight\colon\ weight \in \overline{0, W}$$

$$W_i(weight) = \max_{x_i \in \overline{0, \left[\frac{weight}{weight_i}\right]}} \{x_i \cdot cost_i + W_{i-1}(weight - x_i \cdot weight_i)\}$$

$$\forall\ weight\colon weight \in \overline{0, W} \quad W_0(weight) = 0$$

Количество типов предметов 6, грузоподъемность: 20

5	9	8	7	10	13

Веса

28	20	13	6	21	18

Стоимости

Решение

Шаг №1

$$W_1(0) = \max_{x_i \in \overline{0, \left[\frac{20}{5}\right]}} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$

$$= \max\{0\} = 0, \quad x_1 = 0$$

$$W_1(1) = \max_{x_i \in \overline{0, \left[\frac{20}{5}\right]}} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$

$$= \max\{0\} = 0, \quad x_1 = 0$$

$$W_1(2) = \max_{x_i \in \overline{0, \left[\frac{20}{5}\right]}} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$

$$= \max\{0\} = 0, \quad x_1 = 0$$

$$W_1(3) = \max_{x_i \in \overline{0, \left[\frac{20}{5}\right]}} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$

$$= \max\{0\} = 0, \quad x_1 = 0$$

$$W_1(4) = \max_{x_i \in 0, \left[\frac{20}{5}\right]} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_1 = 0$$
$$W_1(5) = \max_{x_i \in 0, \left[\frac{20}{5}\right]} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0, 28\} = 28, \quad x_1 = 1$$
$$W_1(6) = \max_{x_i \in 0, \left[\frac{20}{5}\right]} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0, 28\} = 28, \quad x_1 = 1$$
$$W_1(7) = \max_{x_i \in 0, \left[\frac{20}{5}\right]} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0, 28\} = 28, \quad x_1 = 1$$
$$W_1(8) = \max_{x_i \in 0, \left[\frac{20}{5}\right]} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0, 28\} = 28, \quad x_1 = 1$$
$$W_1(9) = \max_{x_i \in 0, \left[\frac{20}{5}\right]} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0, 28\} = 28, \quad x_1 = 1$$
$$W_1(10) = \max_{x_i \in 0, \left[\frac{20}{5}\right]} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0, 28, 56\} = 56, \quad x_1 = 2$$
$$W_1(11) = \max_{x_i \in 0, \left[\frac{20}{5}\right]} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0, 28, 56\} = 56, \quad x_1 = 2$$
$$W_1(12) = \max_{x_i \in 0, \left[\frac{20}{5}\right]} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0, 28, 56\} = 56, \quad x_1 = 2$$
$$W_1(13) = \max_{x_i \in 0, \left[\frac{20}{5}\right]} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0, 28, 56\} = 56, \quad x_1 = 2$$
$$W_1(14) = \max_{x_i \in 0, \left[\frac{20}{5}\right]} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0, 28, 56\} = 56, \quad x_1 = 2$$
$$W_1(15) = \max_{x_i \in 0, \left[\frac{20}{5}\right]} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0, 28, 56, 84\} = 84, \quad x_1 = 3$$
$$W_1(16) = \max_{x_i \in 0, \left[\frac{20}{5}\right]} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0, 28, 56, 84\} = 84, \quad x_1 = 3$$
$$W_1(17) = \max_{x_i \in 0, \left[\frac{20}{5}\right]} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0, 28, 56, 84\} = 84, \quad x_1 = 3$$
$$W_1(18) = \max_{x_i \in 0, \left[\frac{20}{5}\right]} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0, 28, 56, 84\} = 84, \quad x_1 = 3$$

$$W_1(19) = \max_{x_i \in \overline{0,\left[\frac{20}{5}\right]}} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0, 28, 56, 84\} = 84, \quad x_1 = 3$$
$$W_1(20) = \max_{x_i \in \overline{0,\left[\frac{20}{5}\right]}} \{x_i \cdot cost_i + W_{1-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0, 28, 56, 84, 112\} = 112, \quad x_1 = 4$$

Шаг №2

$$W_2(0) = \max_{x_i \in \overline{0,\left[\frac{20}{9}\right]}} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_2 = 0$$
$$W_2(1) = \max_{x_i \in \overline{0,\left[\frac{20}{9}\right]}} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_2 = 0$$
$$W_2(2) = \max_{x_i \in \overline{0,\left[\frac{20}{9}\right]}} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_2 = 0$$
$$W_2(3) = \max_{x_i \in \overline{0,\left[\frac{20}{9}\right]}} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_2 = 0$$
$$W_2(4) = \max_{x_i \in \overline{0,\left[\frac{20}{9}\right]}} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_2 = 0$$
$$W_2(5) = \max_{x_i \in \overline{0,\left[\frac{20}{9}\right]}} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28\} = 28, \quad x_2 = 0$$
$$W_2(6) = \max_{x_i \in \overline{0,\left[\frac{20}{9}\right]}} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28\} = 28, \quad x_2 = 0$$
$$W_2(7) = \max_{x_i \in \overline{0,\left[\frac{20}{9}\right]}} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28\} = 28, \quad x_2 = 0$$
$$W_2(8) = \max_{x_i \in \overline{0,\left[\frac{20}{9}\right]}} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28\} = 28, \quad x_2 = 0$$
$$W_2(9) = \max_{x_i \in \overline{0,\left[\frac{20}{9}\right]}} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28, 20\} = 28, \quad x_2 = 0$$
$$W_2(10) = \max_{x_i \in \overline{0,\left[\frac{20}{9}\right]}} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56, 20\} = 56, \quad x_2 = 0$$
$$W_2(11) = \max_{x_i \in \overline{0,\left[\frac{20}{9}\right]}} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56, 20\} = 56, \quad x_2 = 0$$
$$W_2(12) = \max_{x_i \in \overline{0,\left[\frac{20}{9}\right]}} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56, 20\} = 56, \quad x_2 = 0$$

$$W_2(13) = \max_{x_i \in 0, \left[\frac{20}{9}\right]} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56, 20\} = 56, \quad x_2 = 0$$
$$W_2(14) = \max_{x_i \in 0, \left[\frac{20}{9}\right]} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56, 48\} = 56, \quad x_2 = 0$$
$$W_2(15) = \max_{x_i \in 0, \left[\frac{20}{9}\right]} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{84, 48\} = 84, \quad x_2 = 0$$
$$W_2(16) = \max_{x_i \in 0, \left[\frac{20}{9}\right]} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{84, 48\} = 84, \quad x_2 = 0$$
$$W_2(17) = \max_{x_i \in 0, \left[\frac{20}{9}\right]} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{84, 48\} = 84, \quad x_2 = 0$$
$$W_2(18) = \max_{x_i \in 0, \left[\frac{20}{9}\right]} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{84, 48, 40\} = 84, \quad x_2 = 0$$
$$W_2(19) = \max_{x_i \in 0, \left[\frac{20}{9}\right]} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{84, 76, 40\} = 84, \quad x_2 = 0$$
$$W_2(20) = \max_{x_i \in 0, \left[\frac{20}{9}\right]} \{x_i \cdot cost_i + W_{2-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{112, 76, 40\} = 112, \quad x_2 = 0$$

Шаг №3

$$W_3(0) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_3 = 0$$
$$W_3(1) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_3 = 0$$
$$W_3(2) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_3 = 0$$
$$W_3(3) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_3 = 0$$
$$W_3(4) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_3 = 0$$
$$W_3(5) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28\} = 28, \quad x_3 = 0$$
$$W_3(6) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28\} = 28, \quad x_3 = 0$$

$$W_3(7) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$

$$= \max\{28\} = 28, \quad x_3 = 0$$

$$W_3(8) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$

$$= \max\{28, 13\} = 28, \quad x_3 = 0$$

$$W_3(9) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$

$$= \max\{28, 13\} = 28, \quad x_3 = 0$$

$$W_3(10) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$

$$= \max\{56, 13\} = 56, \quad x_3 = 0$$

$$W_3(11) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$

$$= \max\{56, 13\} = 56, \quad x_3 = 0$$

$$W_3(12) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$

$$= \max\{56, 13\} = 56, \quad x_3 = 0$$

$$W_3(13) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$

$$= \max\{56, 41\} = 56, \quad x_3 = 0$$

$$W_3(14) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$

$$= \max\{56, 41\} = 56, \quad x_3 = 0$$

$$W_3(15) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$

$$= \max\{84, 41\} = 84, \quad x_3 = 0$$

$$W_3(16) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$

$$= \max\{84, 41, 26\} = 84, \quad x_3 = 0$$

$$W_3(17) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$

$$= \max\{84, 41, 26\} = 84, \quad x_3 = 0$$

$$W_3(18) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$

$$= \max\{84, 69, 26\} = 84, \quad x_3 = 0$$

$$W_3(19) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$

$$= \max\{84, 69, 26\} = 84, \quad x_3 = 0$$

$$W_3(20) = \max_{x_i \in 0, \left[\frac{20}{8}\right]} \{x_i \cdot cost_i + W_{3-1}(20 - x_i \cdot weight_i)\} =$$

$$= \max\{112, 69, 26\} = 112, \quad x_3 = 0$$

Шаг №4

$$W_4(0) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_4 = 0$$
$$W_4(1) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_4 = 0$$
$$W_4(2) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_4 = 0$$
$$W_4(3) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_4 = 0$$
$$W_4(4) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_4 = 0$$
$$W_4(5) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28\} = 28, \quad x_4 = 0$$
$$W_4(6) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28\} = 28, \quad x_4 = 0$$
$$W_4(7) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28, 6\} = 28, \quad x_4 = 0$$
$$W_4(8) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28, 6\} = 28, \quad x_4 = 0$$
$$W_4(9) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28, 6\} = 28, \quad x_4 = 0$$
$$W_4(10) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56, 6\} = 56, \quad x_4 = 0$$
$$W_4(11) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56, 6\} = 56, \quad x_4 = 0$$
$$W_4(12) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56, 34\} = 56, \quad x_4 = 0$$
$$W_4(13) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56, 34\} = 56, \quad x_4 = 0$$
$$W_4(14) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56, 34, 12\} = 56, \quad x_4 = 0$$

$$W_4(15) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{84, 34, 12\} = 84, \quad x_4 = 0$$
$$W_4(16) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{84, 34, 12\} = 84, \quad x_4 = 0$$
$$W_4(17) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{84, 62, 12\} = 84, \quad x_4 = 0$$
$$W_4(18) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{84, 62, 12\} = 84, \quad x_4 = 0$$
$$W_4(19) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{84, 62, 40\} = 84, \quad x_4 = 0$$
$$W_4(20) = \max_{x_i \in 0, \left[\frac{20}{7}\right]} \{x_i \cdot cost_i + W_{4-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{112, 62, 40\} = 112, \quad x_4 = 0$$

Шаг №5

$$W_5(0) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_5 = 0$$
$$W_5(1) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_5 = 0$$
$$W_5(2) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_5 = 0$$
$$W_5(3) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_5 = 0$$
$$W_5(4) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_5 = 0$$
$$W_5(5) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28\} = 28, \quad x_5 = 0$$
$$W_5(6) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28\} = 28, \quad x_5 = 0$$
$$W_5(7) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28\} = 28, \quad x_5 = 0$$
$$W_5(8) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28\} = 28, \quad x_5 = 0$$

$$W_5(9) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28\} = 28, \quad x_5 = 0$$
$$W_5(10) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56, 21\} = 56, \quad x_5 = 0$$
$$W_5(11) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56, 21\} = 56, \quad x_5 = 0$$
$$W_5(12) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56, 21\} = 56, \quad x_5 = 0$$
$$W_5(13) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56, 21\} = 56, \quad x_5 = 0$$
$$W_5(14) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56, 21\} = 56, \quad x_5 = 0$$
$$W_5(15) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{84, 49\} = 84, \quad x_5 = 0$$
$$W_5(16) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{84, 49\} = 84, \quad x_5 = 0$$
$$W_5(17) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{84, 49\} = 84, \quad x_5 = 0$$
$$W_5(18) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{84, 49\} = 84, \quad x_5 = 0$$
$$W_5(19) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{84, 49\} = 84, \quad x_5 = 0$$
$$W_5(20) = \max_{x_i \in 0, \left[\frac{20}{10}\right]} \{x_i \cdot cost_i + W_{5-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{112, 77, 42\} = 112, \quad x_5 = 0$$

Шаг №6

$$W_6(0) = \max_{x_i \in 0, \left[\frac{20}{13}\right]} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_6 = 0$$
$$W_6(1) = \max_{x_i \in 0, \left[\frac{20}{13}\right]} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_6 = 0$$
$$W_6(2) = \max_{x_i \in 0, \left[\frac{20}{13}\right]} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_6 = 0$$

$$W_6(3) = \max_{x_i \in 0, \left[\frac{20}{13}\right]} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_6 = 0$$
$$W_6(4) = \max_{x_i \in 0, \left[\frac{20}{13}\right]} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{0\} = 0, \quad x_6 = 0$$
$$W_6(5) = \max_{x_i \in 0, \left[\frac{20}{13}\right]} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28\} = 28, \quad x_6 = 0$$
$$W_6(6) = \max_{x_i \in 0, \left[\frac{20}{13}\right]} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28\} = 28, \quad x_6 = 0$$
$$W_6(7) = \max_{x_i \in 0, \left[\frac{20}{13}\right]} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28\} = 28, \quad x_6 = 0$$
$$W_6(8) = \max_{x_i \in 0, \left[\frac{20}{13}\right]} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28\} = 28, \quad x_6 = 0$$
$$W_6(9) = \max_{x_i \in 0, \left[\frac{20}{13}\right]} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{28\} = 28, \quad x_6 = 0$$
$$W_6(10) = \max_{x_i \in 0, \left[\frac{20}{13}\right]} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56\} = 56, \quad x_6 = 0$$
$$W_6(11) = \max_{x_i \in 0, \left[\frac{20}{13}\right]} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56\} = 56, \quad x_6 = 0$$
$$W_6(12) = \max_{x_i \in 0, \left[\frac{20}{13}\right]} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56\} = 56, \quad x_6 = 0$$
$$W_6(13) = \max_{x_i \in 0, \left[\frac{20}{13}\right]} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56, 18\} = 56, \quad x_6 = 0$$
$$W_6(14) = \max_{x_i \in 0, \left[\frac{20}{13}\right]} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{56, 18\} = 56, \quad x_6 = 0$$
$$W_6(15) = \max_{x_i \in 0, \left[\frac{20}{13}\right]} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{84, 18\} = 84, \quad x_6 = 0$$
$$W_6(16) = \max_{x_i \in 0, \left[\frac{20}{13}\right]} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{84, 18\} = 84, \quad x_6 = 0$$
$$W_6(17) = \max_{x_i \in 0, \left[\frac{20}{13}\right]} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} =$$
$$= \max\{84, 18\} = 84, \quad x_6 = 0$$

$$
\begin{aligned}
& W_6(18)=\max_{x_i \in \overline{0, \left[\frac{20}{13}\right]}} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} = \\
& = \max \{84, 46\} = 84, \quad x_6 = 0 \\
& W_6(19)=\max_{x_i \in \overline{0, \left[\frac{20}{13}\right]}} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} = \\
& = \max \{84, 46\} = 84, \quad x_6 = 0 \\
& W_6(20)=\max_{x_i \in \overline{0, \left[\frac{20}{13}\right]}} \{x_i \cdot cost_i + W_{6-1}(20 - x_i \cdot weight_i)\} = \\
& = \max \{112, 46\} = 112, \quad x_6 = 0
\end{aligned}
$$

Ответ

Максимальная стоимость: 112

$$W_6(20) \Rightarrow x_6^o=0$$

$$W_5(20 - 0 * 13)=112 \text{ при } x_5^o=0$$

$$W_4(20 - 0 * 10)=112 \text{ при } x_4^o=0$$

$$W_3(20 - 0 * 7)=112 \text{ при } x_3^o=0$$

$$W_2(20 - 0 * 8)=112 \text{ при } x_2^o=0$$

$$W_1(20 - 0 * 9)=112 \text{ при } x_1^o=4$$

Оптимальное решение: (4, 0, 0, 0, 0, 0)

```
def knapsack(n: int, w: int, weight: list[int], cost: list[int]):
    dp = [[0] * (w + 1) for _ in range(n + 1)]
    x_o = [[0] * (w + 1) for _ in range(n + 1)]
    for type_number in range(1, n + 1):
        print(f"Шаг №{type_number}:")
        for C in range(w + 1):
            print("$$")
            print(r"\large \begin{aligned}")
            print(fr"& W_{type_number}({C})=\max_" + r"{x_i \in \overline{0, \left[\frac{" + str(w)
            print(r"x_i \cdot cost_i + W_{" + f"{type_number}-1" + "}(" + str(w) + r" - x_i \cdot we
            dp[type_number][C] = dp[type_number - 1][C]  # when subject of type_number type is not t
            print(r"& = \max \{")
            print(dp[type_number][C], end='')
            max_items = C // weight[type_number]  # max count of subject of type_number type
            for x in range(1, max_items + 1):
                cur = x * cost[type_number] + dp[type_number - 1][C - x * weight[type_number]]
                print(", " + str(cur), end='')
                if dp[type_number][C] < cur:
                    x_o[type_number][C] = x
                    dp[type_number][C] = cur
            print(r"\}=", rf"{dp[type_number][C]}, \quad x_{type_number} = {x_o[type_number][C]} \\"
            print(r"\end{aligned}")
            print("$$")
        # print_markdown_matrix([[i for i in range(w + 1)]] + [dp[type_number]] + [x_o[type_number]]
    print(f"###### Ответ")
    print(f"Максимальная стоимость: {dp[n][w]}")
    w_ = w
    x_optimal = [str(x_o[n][w_])]
    print(f"$W_{n}({w_})$ при $x_{n}^o={x_o[n][w_]}$")
    for n_ in range(n - 1, 0, -1):
        print(f"$W_{n_}({w_} - {x_o[n_ + 1][w_]} * {weight[n_ + 1]})=", end='')
        w_ -= x_o[n_ + 1][w_] * weight[n_ + 1]
        print(f"{dp[n_][w_]}$ при $x_{n_}^o={x_o[n_][w_]}$")
```

```
        x_optimal.append(str(x_o[n_][w_]))
    print("Оптимальное решение: (" + ", ".join(x_optimal[::-1]) + ")")

def print_markdown_matrix(matrix):
    m = len(matrix[0])
    print()
    print("| " * m + "|")
    print("| --- " * m + "|")
    for i in range(len(matrix)):
        print(f"|", end='')
        for j in range(len(matrix[i])):
            print(matrix[i][j], end='|')
        print()
    print()

def get_problem_from_file(file_name):
    with open(file_name) as f:
        n, w = int(f.readline()), int(f.readline())
        weight = [-1] + list(map(int, f.readline().split()))
        cost = [-1] + list(map(int, f.readline().split()))
    return n, w, weight, cost

n, w, weight, cost = get_problem_from_file("Work10/input2.txt")
print(f"Количество типов предметов {n}, грузоподъемность: {w}")
print("Веса:")
print_markdown_matrix([weight[1:]])
print("Стоимости:")
print_markdown_matrix([cost[1:]])
print("Решение:")
knapsack(n, w, weight, cost)
```