

Министерство науки и высшего образования Российской  
Федерации



Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Московский государственный технический  
университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления»

---

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

---

## Отчет по лабораторной работе №1 по дисциплине «Методы вычислений»

Тема Венгерский метод

---

Студент Золотухин А.В.

---

Группа ИУ7-11М

---

Преподаватель Власов П.А.

---

Москва — 2024 г.

# 1 Задание

## 1.1 Цель работы

**Цель работы:** изучение венгерского метода решения задачи о назначениях.

**Задание:**

1. Реализовать венгерский метод решения задачи о назначениях в виде программы на ЭВМ.
2. Провести решение задачи с матрицей стоимостей, заданной в индивидуальном варианте, рассмотрев два случая:
  - задача о назначениях является задачей минимизации,
  - задача о назначениях является задачей максимизации.

## 2 Теоретическая часть

### 2.1 Содержательная и математическая постановки задачи о назначениях

**Содержательная постановка:** имеется  $n$  работ и  $n$  исполнителей; стоимость выполнения  $i$ -ой работы  $j$ -ым исполнителем составляет  $c_{ij} \geq 0$  единиц. Требуется распределить все работы между исполнителями так, чтобы

- каждый исполнитель выполнял 1 работу;
- общая стоимость выполнения всех работ была  $\min$ .

Введём управляемые переменные:

$$x_{ij} = \begin{cases} 1, & \text{если } i\text{-ую работу выполняет } j\text{-ый работник,} \\ 0, & \text{иначе;} \end{cases} \\ i, j = \overline{1; n}. \quad (2.1)$$

Из переменных  $x_{ij}$ ,  $i, j = \overline{1; n}$ , составим

$$X = (x_{ij}), i, j = \overline{1; n}, \quad (2.2)$$

которую назовём матрицей назначений.

Стоимости выполнения работ также записываем в матрицу

$$C = (c_{ij}), i, j = \overline{1; n}, \quad (2.3)$$

называемой матрицей стоимостей.

Тогда:

1. Стоимость выполнения работ:

$$f = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}. \quad (2.4)$$

2. Условие того, что  $i$ -ую работу выполнит ровно один работник:

$$\sum_{j=1}^n x_{ij} = 1, \quad i = \overline{1; n}. \quad (2.5)$$

3. Условие того, что  $j$ -ый работник выполнит ровно одну работу:

$$\sum_{i=1}^n x_{ij} = 1, \quad j = \overline{1; n}. \quad (2.6)$$

Таким образом приходим к **математической постановке**:

$$\begin{cases} f = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min, \\ \sum_{j=1}^n x_{ij} = 1, \quad i = \overline{1; n}, \\ \sum_{i=1}^n x_{ij} = 1, \quad j = \overline{1; n}, \\ x_{ij} \in \{0, 1\}, \quad i, j = \overline{1; n}. \end{cases} \quad (2.7)$$

## 2.2 Исходные данные варианта №5

$$C = \begin{bmatrix} 9 & 11 & 3 & 6 & 6 \\ 10 & 9 & 11 & 5 & 6 \\ 8 & 10 & 5 & 6 & 4 \\ 6 & 8 & 10 & 4 & 9 \\ 11 & 10 & 9 & 8 & 7 \end{bmatrix} \quad (2.8)$$

## 2.3 Краткое описание венгерского метода

Схема венгерского метода решения задачи о назначениях представлена на рисунке 2.1.

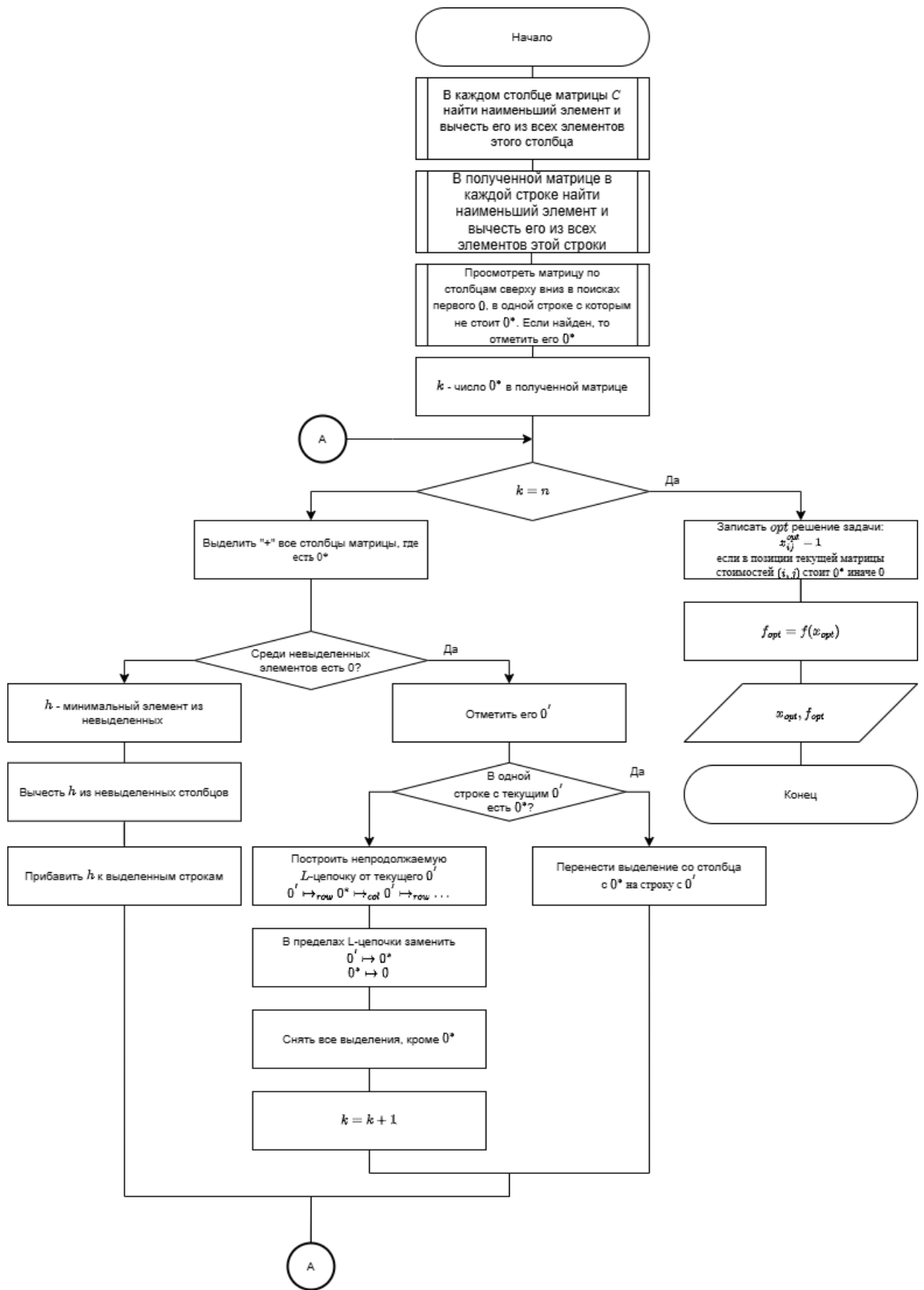


Рисунок 2.1 – Схема венгерского метода

### 3 Практическая часть

```
1 function lab1()  
2     clc;  
3     debugMode = 1;  
4     findMax = 0;  
5  
6     matr = [  
7         9    11    3    6    6;  
8         10    9    11    5    6;  
9         8    10    5    6    4;  
10        6    8    10    4    9;  
11        11   10    9    8    7];  
12     disp('5_вариант. Матрица: ');  
13     disp(matr);  
14  
15     C = matr;  
16  
17     if findMax == 1  
18         C = convertToMin(matr);  
19     end  
20  
21     if debugMode == 1  
22         disp('Матрица_после_приведения_к_задаче_минимизации: ');  
23         disp(C);  
24     end  
25  
26     C = SubtractMinFromCols(C);  
27     if (debugMode == 1)  
28         disp('Вычесть_минимум_из_каждого_столбца');  
29         disp(C);  
30     end  
31  
32     C = SubtractMinFromRows(C);  
33     if (debugMode == 1)  
34         disp('Вычесть_минимум_из_каждой_строки');  
35         disp(C);  
36     end  
37  
38     snn = getSNN_Init(C);  
39     k = length(find(snn(:,2) > 0));
```

```

40     if debugMode == 1
41         disp('Начальная СМН');
42         print_SNN(C, snn);
43     end
44     if debugMode == 1
45         fprintf('Число нулей в построенной СМН: k=%d\n\n', k);
46     end
47     [r, c] = size(C);
48
49     iteration = 1;
50     while k < c
51         if debugMode == 1
52             fprintf('_____Итерация №%d\n\n', iteration);
53         end
54
55         shtrih = zeros(r * c / 2, 2); % позиции 0'
56         b = 1;
57         selectedColumns = snn(:, 2);
58         selectedRows = zeros(1, r);
59         selection = getSelection(r, c, selectedColumns);
60         if debugMode == 1
61             disp('Результат выделения столбцов, в которых стоит 0*');
62             printMarkedMatr(C, snn, shtrih, selectedColumns,
63                             selectedRows);
64         end
65         flag = true;
66         shp = [-1 -1];
67         while flag
68             if debugMode == 1
69                 disp('Поиск 0 среди невыделенных элементов');
70             end
71
72             shp = findShtrih(C, selection);
73             if shp(1) == -1
74                 C = updateMatrNoZero(C, r, c, selection,
75                                     selectedRows, selectedColumns);
76                 if debugMode == 1
77                     disp('Т.к. среди невыделенных элементов нет нулей, матрица была преобразована');
78                 end
79             end
80         end
81     end
82     k = k + 1;
83     snn = C;
84     iteration = iteration + 1;
85 end

```

```

76         printMarkedMatr(C, snn, shtrih,
77                         selectedColumns, selectedRows);
78     end
79     shp = findShtrih(C, selection);
80 end
81 shtrih(b,:) = shp;
82 b = b+1;
83 if debugMode == 1
84     disp('Матрица найденным 0-штрих');
85     printMarkedMatr(C, snn, shtrih, selectedColumns,
86                     selectedRows);
87 end
88 zeroStarInRow = getZeroStarInRow(shp, snn);
89 if isempty(zeroStarInRow)
90     flag = false;
91 else
92     selection(:, zeroStarInRow(2)) = selection(:,
93         zeroStarInRow(2)) - 1;
94     selectedColumns(zeroStarInRow(2)) = 0;
95     selection(zeroStarInRow(1), :) =
96         selection(zeroStarInRow(1), :) + 1;
97     selectedRows(zeroStarInRow(1)) = 1;
98     if debugMode == 1
99         disp('Тк... в одной строке 0-штрих есть 0*,
100             было переброшено выделение: ');
101         printMarkedMatr(C, snn, shtrih,
102                         selectedColumns, selectedRows);
103     end
104 end
105 end
106
107 if debugMode == 1
108     disp('Лцепочка-: ');
109 end
110
111 [shtrih, snn] = createL(shp, shtrih, snn);
112
113 k = length(find(snn(:,2) > 0));

```



```

111         if debugMode == 1
112             disp('Текущая ЧНН: ');
113             print_SNN(C, snn);
114             fprintf('Итого, k=%d\n', k);
115         end
116
117         iteration = iteration + 1;
118         disp('_____');
119     end
120
121     disp('Конечная ЧНН: ');
122     print_SNN(C, snn);
123
124     disp('X=');
125     print_X(snn);
126
127     fOpt = getFOpt(matr, snn);
128     fprintf('Результат(" = %d\n", fOpt);
129 end
130
131 function matr = convertToMin(matr)
132     maxElem = max(matr, [], 'all');
133     matr = (-1) * matr + maxElem;
134 end
135
136 function matr = SubtractMinFromCols(matr)
137     minElemArr = min(matr);
138     for i = 1 : length(matr)
139         matr(:, i) = matr(:, i) - minElemArr(i);
140     end
141 end
142
143 function matr = SubtractMinFromRows(matr)
144     minElemArr = min(matr, [], 2);
145     for i = 1 : length(minElemArr)
146         matr(i, :) = matr(i, :) - minElemArr(i);
147     end
148 end
149
150 function [m, n] = getSNN_Init(matr)
151     [m, n] = size(matr);

```

```

152     SNN = zeros(n, 2);
153     for i = 1: n
154         for j = 1 : m
155             if matr(j, i) == 0
156                 k = 1;
157                 while SNN(k, 1) ~= j && SNN(k, 2) ~= i && k < n
158                     k = k + 1;
159                 end
160                 if (k == n)
161                     SNN(i, 1) = j;
162                     SNN(i, 2) = i;
163                 end
164             end
165         end
166     end
167 end
168
169 function [] = print_SNN(matr, SNN)
170     [r, c] = size(matr);
171     fprintf("\n");
172     for i = 1 : r
173         for j = 1 : c
174             inds = [i, j];
175             f = find(ismember(SNN, inds, "rows"), 1);
176             if (f > 0)
177                 fprintf("\t%d*", matr(i, j));
178             else
179                 fprintf("\t%d", matr(i, j));
180             end
181         end
182         fprintf("\n");
183     end
184     fprintf("\n");
185 end
186
187 function [] = print_X(SNN)
188     [r, ~] = size(SNN);
189     fprintf("\n");
190     for i = 1 : r
191         for j = 1 : r
192             inds = [i, j];

```

```

193         f = find(ismember(SNN,inds , "rows"), 1);
194         if (f > 0)
195             fprintf("\t1");
196         else
197             fprintf("\t0");
198         end
199     end
200     fprintf("\n");
201 end
202 fprintf("\n");
203 end
204
205 function [selection] = getSelection(r, c, selectedColumns)
206     selection = zeros(r, c);
207     for i = 1 : c
208         if selectedColumns(i) > 0
209             selection(:, i) = selection(:, i) + 1;
210         end
211     end
212 end
213
214 function [] = printMarkedMatr(matr, SNN, shtrih , selectedCols ,
    selectedRows)
215     [r,c] = size(matr);
216
217     for i = 1 : r
218         if selectedRows(i) > 0
219             fprintf("+")
220         end
221         for j = 1 : c
222             fprintf("\t%d", matr(i, j));
223             inds = [i, j];
224             f1 = find(ismember(SNN, inds , "rows"), 1);
225             f2 = find(ismember(shtrih , inds , "rows"), 1);
226             if (f1 > 0)
227                 fprintf("*");
228             elseif(f2 > 0)
229                 fprintf(" ")
230             end
231         end
232     end
    fprintf('\n');

```

```

233     end
234
235     for i = 1 : c
236         if selectedCols(i) > 0
237             fprintf("\t+")
238         else
239             fprintf(" \t")
240         end
241     end
242     fprintf('\n\n');
243 end
244
245 function [shp] = findShtrih(matr, selection)
246     shp = [-1 -1];
247     [r, c] = size(matr);
248     for i = 1 : c
249         for j = 1 : r
250             if selection(j, i) == 0 && matr(j, i) == 0
251                 shp = [j, i];
252                 return;
253             end
254         end
255     end
256 end
257
258 function [matr] = updateMatrNoZero(matr, r, c, selection,
    selectedRows, selectedColumns)
259     h = 1e5;
260     for i = 1 : c
261         for j = 1 : r
262             if selection(j, i) == 0 && matr(j, i) < h
263                 h = matr(j, i);
264             end
265         end
266     end
267
268     for i = 1 : c
269         if selectedColumns(i) == 0
270             matr(:, i) = matr(:, i) - h;
271         end
272     end

```

```

273     for i = 1 : r
274         if selectedRows(i) > 0
275             matr(i, :) = matr(i, :) + h;
276         end
277     end
278 end
279
280 function [zeroStarInRow] = getZeroStarInRow(shp, snn)
281     j = shp(1);
282     i = snn(:,1)==j;
283     zeroStarInRow = snn(i, :);
284 end
285
286 function [shtrih, snn] = createL(shp, shtrih, snn)
287     i = shp(1);
288     j = shp(2);
289     fprintf("[%d, %d]\n", i, j);
290     while true
291         inds = [i, j];
292         k = ismember(shtrih, inds, "rows");
293         shtrih(k, :) = [0, 0];
294         k = snn(:,2)==j;
295
296         if k==0
297             snn(j,:) = inds;
298             break;
299         end
300         s = snn(k,:);
301         fprintf("[%d, %d] -> ", s(1), s(2));
302         snn(j,:) = inds;
303         k = shtrih(:,1)==s(2);
304         inds = shtrih(k,:);
305         i = inds(1);
306         j = inds(2);
307         fprintf("[%d, %d]", i, j);
308         fprintf("\n");
309     end
310 end
311
312 function [fOpt] = getFOpt(matr, snn)
313     fOpt = 0;

```

```

314     [r, ~] = size(snn);
315
316     for i = 1 : r
317         fOpt = fOpt + matr(snn(i, 1), snn(i, 2));
318     end
319 end

```

## 3.1 Результаты расчетов

Решение задачи минимизации.

5 вариант. Матрица:

9	11	3	6	6
10	9	11	5	6
8	10	5	6	4
6	8	10	4	9
11	10	9	8	7

Матрица после приведения к задаче минимизации:

9	11	3	6	6
10	9	11	5	6
8	10	5	6	4
6	8	10	4	9
11	10	9	8	7

Вычесть минимум из каждого столбца

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

Вычестъ минимум из каждой строки

3	3	0	2	2
3	0	7	0	1
2	2	2	2	0
0	0	7	0	5
3	0	4	2	1

Начальная СНН

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

Число нулей в построенной СНН:  $k = 4$

----- Итерация №1 -----

Результат выделения столбцов, в которых стоит 0\*:

3	3	0*	2	2
3	0*	7	0	1
2	2	2	2	0*
0*	0	7	0	5
3	0	4	2	1
+	+	+		+

Поиск 0 среди невыделенных элементов

Матрица с найденным 0-штрих

3	3	0*	2	2
3	0*	7	0'	1
2	2	2	2	0*
0*	0	7	0	5

3	0	4	2	1
+	+	+		+

Т.к. в одной строке с 0-штрих есть 0\*, было переброшено выделение:

	3	3	0*	2	2
+	3	0*	7	0'	1
	2	2	2	2	0*
	0*	0	7	0	5
	3	0	4	2	1
	+		+		+

Поиск 0 среди невыделенных элементов

Матрица с найденным 0-штрих

	3	3	0*	2	2
+	3	0*	7	0'	1
	2	2	2	2	0*
	0*	0'	7	0	5
	3	0	4	2	1
	+		+		+

Т.к. в одной строке с 0-штрих есть 0\*, было переброшено выделение:

	3	3	0*	2	2
+	3	0*	7	0'	1
	2	2	2	2	0*
+	0*	0'	7	0	5
	3	0	4	2	1
		+		+	

Поиск 0 среди невыделенных элементов

Матрица с найденным 0-штрих

	3	3	0*	2	2
+	3	0*	7	0'	1
	2	2	2	2	0*
+	0*	0'	7	0	5



$$\begin{array}{ccccc} 3 & 0' & 4 & 2 & 1 \\ & + & & + & \end{array}$$

L-цепочка:

[5, 2]

[2, 2]  $\rightarrow$  [2, 4]

Текущая СНН:

$$\begin{array}{ccccc} 3 & 3 & 0* & 2 & 2 \\ 3 & 0 & 7 & 0* & 1 \\ 2 & 2 & 2 & 2 & 0* \\ 0* & 0 & 7 & 0 & 5 \\ 3 & 0* & 4 & 2 & 1 \end{array}$$

Итого,  $k = 5$

-----

Конечная СНН:

$$\begin{array}{ccccc} 3 & 3 & 0* & 2 & 2 \\ 3 & 0 & 7 & 0* & 1 \\ 2 & 2 & 2 & 2 & 0* \\ 0* & 0 & 7 & 0 & 5 \\ 3 & 0* & 4 & 2 & 1 \end{array}$$

X =

$$\begin{array}{ccccc} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{array}$$

Результат = 28

Решение задачи максимизации.

5 вариант. Матрица:

9	11	3	6	6
10	9	11	5	6
8	10	5	6	4
6	8	10	4	9
11	10	9	8	7

Матрица после приведения к задаче минимизации:

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

Вычесть минимум из каждого столбца

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

Вычесть минимум из каждой строки

2	0	8	2	3
1	2	0	3	3
2	0	5	1	4
5	3	1	4	0
0	1	2	0	2

Начальная СНН

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

Число нулей в построенной СНН:  $k = 4$

----- Итерация №1 -----

Результат выделения столбцов, в которых стоит 0\*:

2	0*	8	2	3
1	2	0*	3	3
2	0	5	1	4
5	3	1	4	0*
0*	1	2	0	2
+	+	+		+

Поиск 0 среди невыделенных элементов

Матрица с найденным 0-штрих

2	0*	8	2	3
1	2	0*	3	3
2	0	5	1	4
5	3	1	4	0*
0*	1	2	0'	2
+	+	+		+

Т.к. в одной строке с 0-штрих есть 0\*, было переброшено выделение:

	2	0*	8	2	3
	1	2	0*	3	3
	2	0	5	1	4
	5	3	1	4	0*
+	0*	1	2	0'	2
		+	+		+

Поиск 0 среди невыделенных элементов

Т.к. среди невыделенных элементов нет нулей, матрица была преобразована

	1	0*	8	1	3
	0	2	0*	2	3
	1	0	5	0	4
	4	3	1	3	0*
+	0*	2	3	0'	3
		+	+		+

Матрица с найденным 0-штрих

	1	0*	8	1	3
	0'	2	0*	2	3
	1	0	5	0	4
	4	3	1	3	0*
+	0*	2	3	0'	3
		+	+		+

Т.к. в одной строке с 0-штрих есть 0\*, было переброшено выделение:

	1	0*	8	1	3
+	0'	2	0*	2	3
	1	0	5	0	4
	4	3	1	3	0*
+	0*	2	3	0'	3
		+			+

Поиск 0 среди невыделенных элементов

Матрица с найденным 0-штрих

	1	0*	8	1	3
+	0'	2	0*	2	3
	1	0	5	0'	4
	4	3	1	3	0*
+	0*	2	3	0'	3
		+			+

L-цепочка:

[3, 4]

Текущая СНН:

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

Итого,  $k = 5$

-----

Конечная СНН:

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

$X =$

0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1
1	0	0	0	0

Результат = 48