



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 (ИУ)»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии (ИУ7)»

ОТЧЕТ

Лабораторная работа №1

по курсу «Методы вычислений»

на тему: «Венгерский метод решения задачи о назначениях»

Вариант № 9

Студент ИУ7-12М
(Группа)

(Подпись, дата)

К.Э. Ковалец
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

П.А. Власов
(И. О. Фамилия)

2023 г.

1 Теоретическая часть

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.

1.1 Содержательная и математическая постановки задачи о назначениях

Содержательная постановка: имеется n работ и n испытателей; 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;
- каждую работу выполнял только 1 исполнитель;
- общая стоимость выполнения всех работ была \min .

Введем управляемые переменные:

$$x_{ij} = \begin{cases} 1, & \text{если } i\text{-ую работу выполняет } j\text{-ый работник,} \\ 0, & \text{иначе;} \end{cases} \\ i, j = \overline{1; n}. \quad (1.1)$$

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

$$X = (x_{ij})_{i,j=\overline{1;n}}, \quad (1.2)$$

которую назовем матрицей назначений.

Стоимости выполнения работ также записываем в матрицу

$$C = (c_{ij})_{i,j=\overline{1;n}}, \quad (1.3)$$

называемой матрицей стоимостей.

Тогда:

1. Стоимость выполнения работ:

$$f = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}. \quad (1.4)$$

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

$$\sum_{j=1}^n x_{ij} = 1, \quad i = \overline{1;n}. \quad (1.5)$$

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

$$\sum_{i=1}^n x_{ij} = 1, \quad j = \overline{1;n}. \quad (1.6)$$

Таким образом приходим к **математической постановке**:

$$\begin{cases} f = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min, \\ \sum_{j=1}^n x_{ij} = 1, \quad i = \overline{1;n}, \\ \sum_{i=1}^n x_{ij} = 1, \quad j = \overline{1;n}, \\ x_{ij} \in \{0, 1\}, \quad i, j = \overline{1;n}. \end{cases} \quad (1.7)$$

1.2 Исходные данные конкретного варианта

Вариант 9:

$$C = \begin{bmatrix} 4 & 7 & 1 & 5 & 5 \\ 6 & 8 & 3 & 7 & 6 \\ 6 & 4 & 5 & 7 & 7 \\ 4 & 2 & 3 & 4 & 9 \\ 8 & 1 & 8 & 3 & 8 \end{bmatrix} \quad (1.8)$$

1.3 Краткое описание венгерского метода

Схема венгерского метода решения задачи о назначениях представлена на рисунках 1.1–1.2.

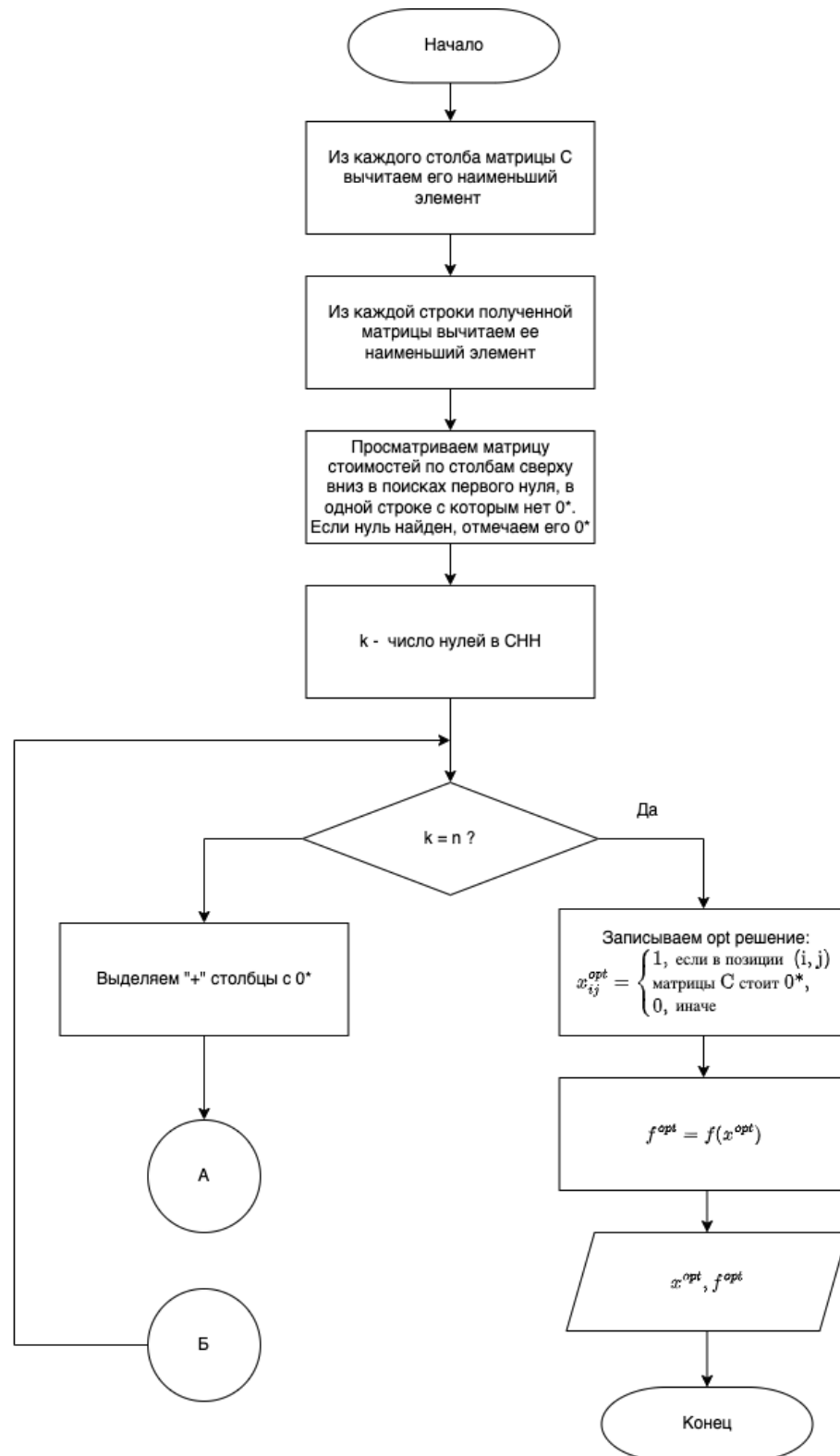


Рисунок 1.1 – Схема венгерского метода решения задачи о назначениях (часть 1)

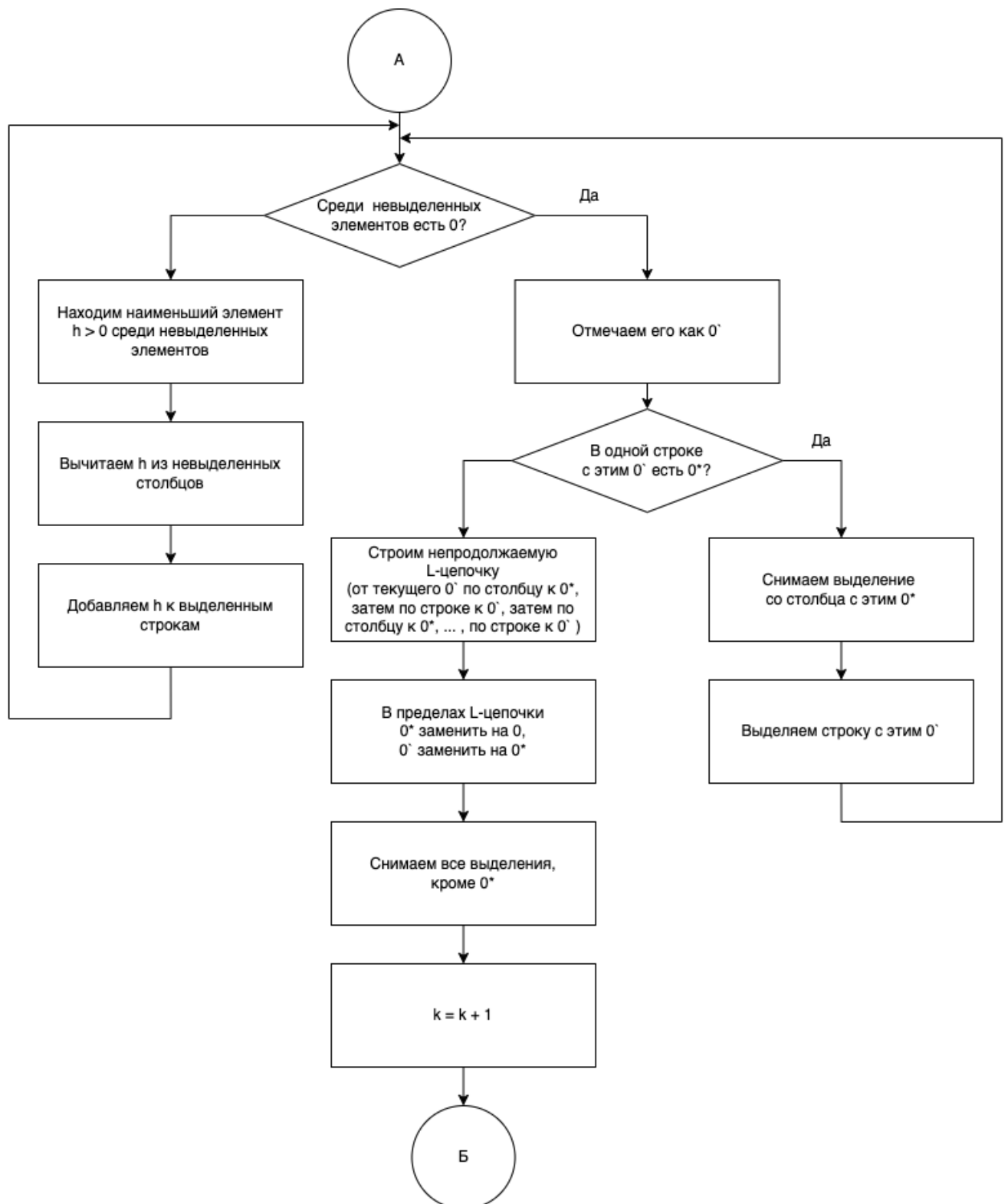


Рисунок 1.2 – Схема венгерского метода решения задачи о назначениях (часть 2)

2 Практическая часть

2.1 Текст программы

В листинге 2.1 представлен код программы.

Листинг 2.1 — Код программы

```
1  function lab_01()
2      clc;
3      findMax = 0;
4      debugFlg = 1;
5      [base_color, red, green, yellow, blue, purple] = getColors();
6
7      matr = [
8          4 7 1 5 5;
9          6 8 3 7 6;
10         6 4 5 7 7;
11         4 2 3 4 9;
12         8 1 8 3 8
13     ];
14
15     fprintf('%sМатрица стоимостей для 9 варианта:%s\n', green, base_color);
16     disp(matr);
17
18     C = matr;
19
20     if findMax == 1
21         C = convertToMin(matr);
22
23         if debugFlg == 1
24             fprintf('%s\nМатрица после приведения к задаче минимизации:%s\n', blue,
↪ base_color);
25             disp(C);
26         end
27     end
28
29     C = updateColumns(C);
30     if debugFlg == 1
31         fprintf('%s\nРезультат вычитания наименьшего элемента по столбцам:%s\n',
↪ blue, base_color);
32         disp(C);
33     end
34
35     C = updateRows(C);
36     if debugFlg == 1
```

Продолжение листинга 2.1

```

37         fprintf('%s\nРезультат вычитания наименьшего элемента по строкам:%s\n',
↪ blue, base_color);
38         disp(C);
39     end
40
41     [numRows, numCols] = size(C);
42
43     matrSIZ = getSIZInit(C);
44     if debugFlg == 1
45         fprintf('%s\nНачальная СНН:%s\n', blue, base_color);
46         printSIZ(C, matrSIZ);
47     end
48
49     k = sum(sum(matrSIZ));
50     if debugFlg == 1
51         fprintf('%s\nЧисло нулей в построенной СНН:%s k = %d\n', blue, base_color,
↪ k);
52     end
53
54     iteration = 1;
55     while k < numCols
56         if debugFlg == 1
57             fprintf('%s\n_----- Итерация №%d
↪ _-----\n%s', green, iteration, base_color);
58         end
59
60         matrStreak = zeros(numRows, numCols); % матрица, в которой отмечаются
↪ позиции 0'
61         selectedColumns = sum(matrSIZ);
62         selectedRows = zeros(numRows);
63         selection = getSelection(numRows, numCols, selectedColumns);
64
65         if debugFlg == 1
66             fprintf('%s\nРезультат выделения столбцов, в которых стоит 0*:%s\n',
↪ blue, base_color);
67             printMarkedMatr(C, matrSIZ, matrStreak, selectedColumns,
↪ selectedRows);
68         end
69
70         flag = true;
71         streakPnt = [-1 -1];
72         while flag
73             streakPnt = findStreak(C, selection);
74             if streakPnt(1) == -1

```

Продолжение листинга 2.1

```

75         C = updateMatrNoZero(C, numRows, numCols, selection, selectedRows,
↪ selectedColumns);
76
77         if debugFlg == 1
78             fprintf('%s\nСреди невыделенных элементов нет 0, преобразуем
↪ матрицу:%s\n', blue, base_color);
79             printMarkedMatr(C, matrSIZ, matrStreak, selectedColumns,
↪ selectedRows);
80             end
81
82             streakPnt = findStreak(C, selection);
83             end
84
85             matrStreak(streakPnt(1), streakPnt(2)) = 1;
86             if debugFlg == 1
87                 fprintf("%s\nСреди невыделенных элементов есть 0, отметим 0':%s\n",
↪ blue, base_color);
88                 printMarkedMatr(C, matrSIZ, matrStreak, selectedColumns,
↪ selectedRows);
89                 end
90
91             zeroStarInRow = getZeroStarInRow(streakPnt, numCols, matrSIZ);
92             if zeroStarInRow(1) == -1
93                 flag = false;
94             else
95                 % снять выделение со столбца с 0*
96                 selection(:, zeroStarInRow(2)) = selection(:, zeroStarInRow(2)) -
↪ 1;
97                 selectedColumns(zeroStarInRow(2)) = 0;
98
99                 % перенести выделение на строку с 0'
100                 selection(zeroStarInRow(1), :) = selection(zeroStarInRow(1), :) +
↪ 1;
101                 selectedRows(zeroStarInRow(1)) = 1;
102                 if debugFlg == 1
103                     fprintf("%s\nВ одной строке с 0' есть 0*, перебросим выделение
↪ со столбца на строку:%s\n", blue, base_color);
104                     printMarkedMatr(C, matrSIZ, matrStreak, selectedColumns,
↪ selectedRows);
105                     end
106                 end
107             end
108
109             [matrStreak, matrSIZ] = createL(numRows, numCols, streakPnt, matrStreak,
↪ matrSIZ, debugFlg);

```


Продолжение листинга 2.1

```

110
111         k = sum(sum(matrSIZ));
112         if debugFlg == 1
113             fprintf("%s\nВ пределах L-цепочки 0* заменим на 0, а 0' на 0*:\n%s",
↪ blue, base_color);
114             printSIZ(C, matrSIZ);
115             fprintf('%s\nЧисло нулей в построенной СНН:%s k = %d\n', blue,
↪ base_color, k);
116         end
117
118         iteration = iteration + 1;
119     end
120
121     if debugFlg == 1
122         fprintf('%s\nКонечная СНН:\n%s', blue, base_color);
123         printSIZ(C, matrSIZ);
124     end
125
126     fprintf('%s\nX =\n%s', green, base_color);
127     disp(matrSIZ);
128
129     fOpt = getFOpt(matr, matrSIZ);
130     fprintf('%s\nf opt = %s%d\n', green, base_color, fOpt);
131 end
132
133 % Найдти первый нулевой элемент среди невыделенных, в одной строке с которым не
↪ стоит 0*
134 function [streakPnt] = findStreak(matr, selection)
135     streakPnt = [-1 -1];
136     [numRows, numCols] = size(matr);
137     for i = 1 : numCols
138         for j = 1 : numRows
139             if selection(j, i) == 0 && matr(j, i) == 0
140                 streakPnt(1) = j;
141                 streakPnt(2) = i;
142                 return;
143             end
144         end
145     end
146 end
147
148 function [] = printSIZ(matr, matrSIZ)
149     [base_color, red, green, yellow, blue, purple] = getColors();
150     [numRows, numCols] = size(matr);
151

```

Продолжение листинга 2.1

```

152     for i = 1 : numRows
153         for j = 1 : numCols
154             if matrSIZ(i, j) == 1
155                 fprintf("%s\t%d*\t%s", purple, matr(i, j), base_color);
156             else
157                 fprintf("\t%d\t", matr(i, j));
158             end
159         end
160         fprintf("\n");
161     end
162 end
163
164 function [] = printMarkedMatr(matr, matrSIZ, matrStreak, selectedCols,
    ↪ selectedRows)
165     [base_color, red, green, yellow, blue, purple] = getColors();
166     [numRows, numCols] = size(matr);
167
168     for i = 1 : numRows
169         if selectedRows(i) == 1
170             fprintf("%s+%s", green, base_color);
171         end
172
173         for j = 1 : numCols
174             if matrSIZ(i, j) == 1
175                 fprintf("%s\t%d*\t%s", purple, matr(i, j), base_color);
176             elseif matrStreak(i, j) == 1
177                 fprintf("%s\t%d'\t%s", red, matr(i, j), base_color);
178             else
179                 fprintf("\t%d\t", matr(i, j));
180             end
181         end
182         fprintf('\n');
183     end
184
185     for i = 1 : numCols
186         if selectedCols(i) == 1
187             fprintf("%s\t+\t%s", green, base_color);
188         else
189             fprintf(" \t\t");
190         end
191     end
192     fprintf('\n');
193 end
194
195 % Приведение задачи максимизации к задаче минимизации

```

Продолжение листинга 2.1

```
196 function matr = convertToMin(matr)
197     maxElem = max(max(matr));
198     matr = matr * (-1) + maxElem;
199 end
200
201 % Нахождение наименьшего элемента в каждом столбце матрицы C
202 % и вычитание его из соответствующего столбца
203 function matr = updateColumns(matr)
204     minElemArr = min(matr);
205     for i = 1 : length(minElemArr)
206         matr(:, i) = matr(:, i) - minElemArr(i);
207     end
208 end
209
210 % Нахождение наименьшего элемента в каждой строке матрицы C
211 % и вычитание его из соответствующей строки
212 function matr = updateRows(matr)
213     minElemArr = min(matr, [], 2);
214     for i = 1 : length(minElemArr)
215         matr(i, :) = matr(i, :) - minElemArr(i);
216     end
217 end
218
219 % Начальное состояние CHH (системы независимых нулей)
220 function matrSIZ = getSIZInit(matr)
221     [numRows, numCols] = size(matr);
222     matrSIZ = zeros(numRows, numCols);
223
224     for i = 1 : numCols
225         for j = 1 : numRows
226             if matr(j, i) == 0
227                 count = 0;
228                 for k = 1 : numCols
229                     count = count + matrSIZ(j, k);
230                 end
231                 for k = 1 : numRows
232                     count = count + matrSIZ(k, i);
233                 end
234                 if count == 0
235                     matrSIZ(j, i) = 1;
236                 end
237             end
238         end
239     end
240 end
```

Продолжение листинга 2.1

```
241
242 % Выделение столбцов, в которых стоит 0*
243 function [selection] = getSelection(numRows, numCols, selectedColumns)
244     selection = zeros(numRows, numCols);
245     for j = 1 : numCols
246         if selectedColumns(j) == 1
247             selection(:, j) = selection(:, j) + 1;
248         end
249     end
250 end
251
252 % Изменить матрицу в случае, если среди невыделенных элементов нет нуля
253 function [matr] = updateMatrNoZero(matr, numRows, numCols, selection, selectedRows,
254     ↪ selectedColumns)
255     h = 1e5; % Наименьший элемент среди невыделенных
256     for i = 1 : numCols
257         for j = 1 : numRows
258             if selection(j, i) == 0 && matr(j, i) < h
259                 h = matr(j, i);
260             end
261         end
262
263         for i = 1 : numCols
264             if selectedColumns(i) == 0
265                 matr(:, i) = matr(:, i) - h;
266             end
267         end
268         for i = 1 : numRows
269             if selectedRows(i) == 1
270                 matr(i, :) = matr(i, :) + h;
271             end
272         end
273     end
274
275 % Найдти 0* в той же строке, что и 0'
276 function [zeroStarInRow] = getZeroStarInRow(streakPnt, numCols, matrSIZ)
277     j = streakPnt(1);
278     zeroStarInRow = [-1 -1];
279     for i = 1 : numCols
280         if matrSIZ(j, i) == 1
281             zeroStarInRow(1) = j;
282             zeroStarInRow(2) = i;
283             break
284         end
285     end
```

Продолжение листинга 2.1

```
285     end
286 end
287
288 % Построить L-цепочку
289 function [matrStreak, matrSIZ] = createL(numRows, numCols, streakPnt, matrStreak,
↪ matrSIZ, debugFlg)
290     [base_color, red, green, yellow, blue, purple] = getColors();
291
292     if debugFlg == 1
293         fprintf('%s\nПостроенная L-цепочка: %s', blue, base_color);
294     end
295
296     i = streakPnt(1);
297     j = streakPnt(2);
298     while i > 0 && j > 0 && i <= numRows && j <= numCols
299         % Снять *
300         matrStreak(i, j) = 0;
301         % Заменить ' на *
302         matrSIZ(i, j) = 1;
303
304         if debugFlg == 1
305             fprintf("[%d, %d] ", i, j);
306         end
307
308         % Дойти до 0* по столбцу от 0'
309         kRow = 1;
310         while kRow <= numRows && (matrSIZ(kRow, j) ~= 1 || kRow == i)
311             kRow = kRow + 1;
312         end
313
314         if (kRow <= numRows)
315             % Дойти до 0' по строке от 0*
316             lCol = 1;
317             while lCol <= numCols && (matrStreak(kRow, lCol) ~= 1 || lCol == j)
318                 lCol = lCol + 1;
319             end
320
321             if lCol <= numCols
322                 matrSIZ(kRow, j) = 0;
323
324                 if debugFlg == 1
325                     fprintf("-> [%d, %d] -> ", kRow, j);
326                 end
327             end
328             j = lCol;
```

Продолжение листинга 2.1

```
329         end
330         i = kRow;
331     end
332
333     if debugFlg == 1
334         fprintf('\n');
335     end
336 end
337
338 function [fOpt] = getFOpt(matr, matrSIZ)
339     fOpt = 0;
340     [numRows, numCols] = size(matr);
341
342     for i = 1 : numCols
343         for j = 1 : numRows
344             if matrSIZ(j, i) == 1
345                 fOpt = fOpt + matr(j, i);
346             end
347         end
348     end
349 end
350
351 function [base_color, red, green, yellow, blue, purple] = getColors()
352     base_color = "\x1B[0m";
353     red        = "\x1B[31m";
354     green      = "\x1B[32m";
355     yellow     = "\x1B[33m";
356     blue       = "\x1B[34m";
357     purple     = "\x1B[35m";
358 end
```

2.2 Результаты расчетов для задач из индивидуального варианта

В листинге 2.2 представлены расчеты для задачи минимизации.

Листинг 2.2 — Задача минимизации

```
1  Матрица стоимостей для 9 варианта:
2      4   7   1   5   5
3      6   8   3   7   6
4      6   4   5   7   7
5      4   2   3   4   9
6      8   1   8   3   8
7
8  Результат вычитания наименьшего элемента по столбцам:
9      0   6   0   2   0
10     2   7   2   4   1
11     2   3   4   4   2
12     0   1   2   1   4
13     4   0   7   0   3
14
15 Результат вычитания наименьшего элемента по строкам:
16     0   6   0   2   0
17     1   6   1   3   0
18     0   1   2   2   0
19     0   1   2   1   4
20     4   0   7   0   3
21
22 Начальная СНН:
23     0*           6           0           2           0
24     1           6           1           3          0*
25     0           1           2           2           0
26     0           1           2           1           4
27     4           0*          7           0           3
28
29 Число нулей в построенной СНН: k = 3
30
31 ----- Итерация №1 -----
32
33 Результат выделения столбцов, в которых стоит 0*:
34     0*           6           0           2           0
35     1           6           1           3          0*
36     0           1           2           2           0
37     0           1           2           1           4
38     4           0*          7           0           3
39     +           +
40
```

Продолжение листинга 2.2

```

41 Среди невыделенных элементов есть 0, отметим 0':
42     0*        6        0'        2        0
43     1        6        1        3        0*
44     0        1        2        2        0
45     0        1        2        1        4
46     4        0*       7        0        3
47     +        +
48
49 В одной строке с 0' есть 0*, перебросим выделение со столбца на строку:
50 +     0*        6        0'        2        0
51     1        6        1        3        0*
52     0        1        2        2        0
53     0        1        2        1        4
54     4        0*       7        0        3
55         +
56
57 Среди невыделенных элементов есть 0, отметим 0':
58 +     0*        6        0'        2        0
59     1        6        1        3        0*
60     0'       1        2        2        0
61     0        1        2        1        4
62     4        0*       7        0        3
63         +
64
65 Построенная L-цепочка: [3, 1] -> [1, 1] -> [1, 3]
66
67 В пределах L-цепочки 0* заменим на 0, а 0' на 0*:
68     0        6        0*        2        0
69     1        6        1        3        0*
70     0*       1        2        2        0
71     0        1        2        1        4
72     4        0*       7        0        3
73
74 Число нулей в построенной СНН: k = 4
75
76 ----- Итерация №2 -----
77
78 Результат выделения столбцов, в которых стоит 0*:
79     0        6        0*        2        0
80     1        6        1        3        0*
81     0*       1        2        2        0
82     0        1        2        1        4
83     4        0*       7        0        3
84     +        +        +
85

```


Продолжение листинга 2.2

```

86  Среди невыделенных элементов есть 0, отметим 0':
87      0          6          0*          2          0
88      1          6          1          3          0*
89      0*         1          2          2          0
90      0          1          2          1          4
91      4          0*         7          0'         3
92      +          +          +          +          +
93
94  В одной строке с 0' есть 0*, перебросим выделение со столбца на строку:
95      0          6          0*          2          0
96      1          6          1          3          0*
97      0*         1          2          2          0
98      0          1          2          1          4
99  +    4          0*         7          0'         3
100     +          +          +          +          +
101
102 Среди невыделенных элементов нет 0, преобразуем матрицу:
103      0          5          0*          1          0
104      1          5          1          2          0*
105      0*         0          2          1          0
106      0          0          2          0          4
107  +    5          0*         8          0'         4
108     +          +          +          +          +
109
110 Среди невыделенных элементов есть 0, отметим 0':
111      0          5          0*          1          0
112      1          5          1          2          0*
113      0*         0'         2          1          0
114      0          0          2          0          4
115  +    5          0*         8          0'         4
116     +          +          +          +          +
117
118 В одной строке с 0' есть 0*, перебросим выделение со столбца на строку:
119      0          5          0*          1          0
120      1          5          1          2          0*
121  +    0*         0'         2          1          0
122      0          0          2          0          4
123  +    5          0*         8          0'         4
124     +          +          +          +          +
125
126 Среди невыделенных элементов есть 0, отметим 0':
127      0'         5          0*          1          0
128      1          5          1          2          0*
129  +    0*         0'         2          1          0
130      0          0          2          0          4

```

Продолжение листинга 2.2

```

131 +      5      0*      8      0'      4
132      +      +
133
134 В одной строке с 0' есть 0*, перебросим выделение со столбца на строку:
135 +      0'      5      0*      1      0
136      1      5      1      2      0*
137 +      0*      0'      2      1      0
138      0      0      2      0      4
139 +      5      0*      8      0'      4
140      +
141
142 Среди невыделенных элементов есть 0, отметим 0':
143 +      0'      5      0*      1      0
144      1      5      1      2      0*
145 +      0*      0'      2      1      0
146      0'      0      2      0      4
147 +      5      0*      8      0'      4
148      +
149
150 Построенная L-цепочка: [4, 1] -> [3, 1] -> [3, 2] -> [5, 2] -> [5, 4]
151
152 В пределах L-цепочки 0* заменим на 0, а 0' на 0*:
153      0      5      0*      1      0
154      1      5      1      2      0*
155      0      0*      2      1      0
156      0*      0      2      0      4
157      5      0      8      0*      4
158
159 Число нулей в построенной СНН: k = 5
160
161 Конечная СНН:
162      0      5      0*      1      0
163      1      5      1      2      0*
164      0      0*      2      1      0
165      0*      0      2      0      4
166      5      0      8      0*      4
167
168 X =
169      0  0  1  0  0
170      0  0  0  0  1
171      0  1  0  0  0
172      1  0  0  0  0
173      0  0  0  1  0
174
175 f opt = 18

```

В листинге 2.3 представлены расчеты для задачи максимизации.

Листинг 2.3 — Задача максимизации

```
1  Матрица стоимостей для 9 варианта:
2      4   7   1   5   5
3      6   8   3   7   6
4      6   4   5   7   7
5      4   2   3   4   9
6      8   1   8   3   8
7
8  Матрица после приведения к задаче минимизации:
9      5   2   8   4   4
10     3   1   6   2   3
11     3   5   4   2   2
12     5   7   6   5   0
13     1   8   1   6   1
14
15 Результат вычитания наименьшего элемента по столбцам:
16     4   1   7   2   4
17     2   0   5   0   3
18     2   4   3   0   2
19     4   6   5   3   0
20     0   7   0   4   1
21
22 Результат вычитания наименьшего элемента по строкам:
23     3   0   6   1   3
24     2   0   5   0   3
25     2   4   3   0   2
26     4   6   5   3   0
27     0   7   0   4   1
28
29 Начальная СНН:
30         3           0*           6           1           3
31         2           0           5           0*          3
32         2           4           3           0           2
33         4           6           5           3          0*
34         0*          7           0           4           1
35
36 Число нулей в построенной СНН: k = 4
37
38 ----- Итерация №1 -----
39
40 Результат выделения столбцов, в которых стоит 0*:
41         3           0*           6           1           3
42         2           0           5           0*          3
43         2           4           3           0           2
```

Продолжение листинга 2.3

```

44         4           6           5           3           0*
45         0*         7           0           4           1
46         +         +           +           +           +
47
48     Среди невыделенных элементов есть 0, отметим 0':
49         3           0*         6           1           3
50         2           0           5           0*         3
51         2           4           3           0           2
52         4           6           5           3           0*
53         0*         7           0'         4           1
54         +         +           +           +           +
55
56     В одной строке с 0' есть 0*, перебросим выделение со столбца на строку:
57         3           0*         6           1           3
58         2           0           5           0*         3
59         2           4           3           0           2
60         4           6           5           3           0*
61     +         0*         7           0'         4           1
62             +           +           +           +
63
64     Среди невыделенных элементов нет 0, преобразуем матрицу:
65         1           0*         4           1           3
66         0           0           3           0*         3
67         0           4           1           0           2
68         2           6           3           3           0*
69     +         0*         9           0'         6           3
70             +           +           +           +
71
72     Среди невыделенных элементов есть 0, отметим 0':
73         1           0*         4           1           3
74         0'         0           3           0*         3
75         0           4           1           0           2
76         2           6           3           3           0*
77     +         0*         9           0'         6           3
78             +           +           +           +
79
80     В одной строке с 0' есть 0*, перебросим выделение со столбца на строку:
81         1           0*         4           1           3
82     +         0'         0           3           0*         3
83         0           4           1           0           2
84         2           6           3           3           0*
85     +         0*         9           0'         6           3
86             +           +           +           +
87
88     Среди невыделенных элементов есть 0, отметим 0':

```

Продолжение листинга 2.3

```

89         1          0*          4          1          3
90   +    0'          0          3          0*          3
91         0'          4          1          0          2
92         2          6          3          3          0*
93   +    0*          9          0'          6          3
94         +          +          +          +          +
95
96   Построенная L-цепочка: [3, 1] -> [5, 1] -> [5, 3]
97
98   В пределах L-цепочки 0* заменим на 0, а 0' на 0*:
99         1          0*          4          1          3
100        0          0          3          0*          3
101        0*         4          1          0          2
102        2          6          3          3          0*
103        0          9          0*         6          3
104
105   Число нулей в построенной СНН: k = 5
106
107   Конечная СНН:
108         1          0*          4          1          3
109        0          0          3          0*          3
110        0*         4          1          0          2
111        2          6          3          3          0*
112        0          9          0*         6          3
113
114   X =
115     0   1   0   0   0
116     0   0   0   1   0
117     1   0   0   0   0
118     0   0   0   0   1
119     0   0   1   0   0
120
121   f opt = 37

```