



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 курсу «Методы вычислений»  
на тему: «Венгерский метод решения»  
Вариант № 15

Студент ИУ7-11М  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Миронов Г. А.  
(И. О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Власов П. А.  
(И. О. Фамилия)

2023 г.

# 1 Теоретические сведения

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

## 1.1 Содержательная постановка задачи

В распоряжении работодателя имеется  $n$  работ и такое же число исполнителей. Стоимость выполнения  $i$ -ой работы  $j$ -ым исполнителем составляет  $c_{ij} \geq 0$  единиц.

Требуется распределить все работы между исполнителями так, чтобы:

1. каждый из них выполнял ровно 1 работу
2. общая стоимость работ была минимальна

## 1.2 Математическая постановка задачи

Введем управляемые переменные:

$$x_{ij} = \begin{cases} 1, & \text{если } i\text{-ую работу выполняет } j\text{-ый исполнитель} \\ 0, & \text{иначе} \end{cases},$$

где  $i, j = \overline{1, n}$ .

Тогда:

1. Общая стоимость всех работ:

$$f = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

2. Условие того, что  $j$ -ый исполнитель выполняет ровно 1 работу:

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

3. Условие того, что  $i$ -ую работу выполняет ровно 1 исполнитель:

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

Таким образом, математическая постановка задачи о назначениях:

$$\left\{ \begin{array}{l} f = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min \\ \sum_{i=1}^n x_{ij} = 1, j = \overline{1, n} \\ \sum_{j=1}^n x_{ij} = 1, i = \overline{1, n} \\ x_{ij} \in \{0, 1\}, i, j = \overline{1, n} \end{array} \right.$$

### 1.3 Схема алгоритма

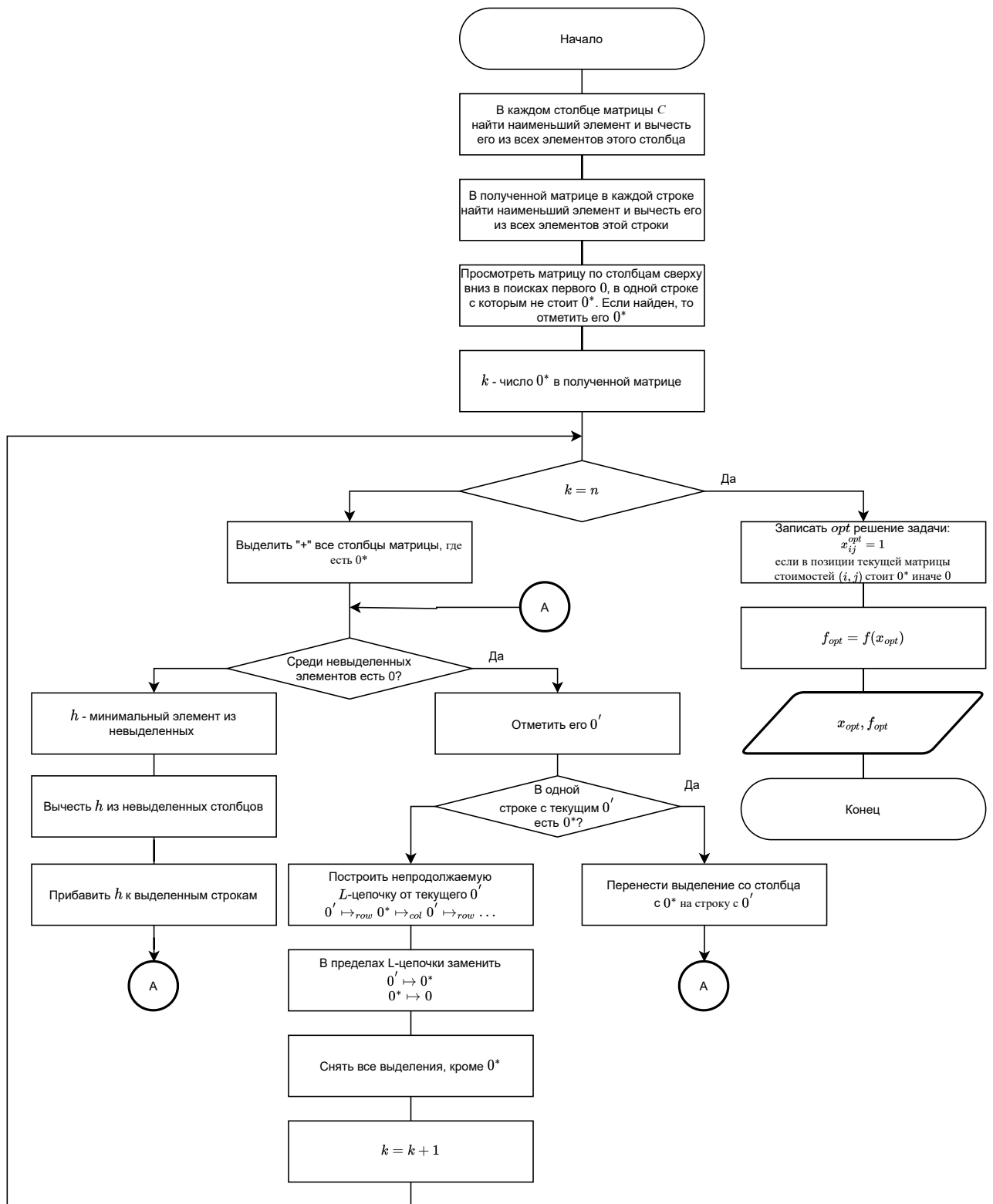


Рисунок 1.1 – Схема алгоритма

## 2 Выполнение индивидуального задания

### 2.1 Постановка задачи

Необходимо:

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:
  - (a) задача о назначениях является задачей минимизации,
  - (b) задача о назначениях является задачей максимизации.

Индивидуальный вариант матрицы стоимостей:

$$C = \begin{bmatrix} 10 & 12 & 7 & 11 & 10 \\ 12 & 5 & 12 & 7 & 12 \\ 8 & 6 & 7 & 8 & 13 \\ 8 & 11 & 5 & 9 & 9 \\ 10 & 8 & 9 & 11 & 11 \end{bmatrix}$$

### 2.2 Текст программы

Листинг 2.1 – Файл main.m.

```
function main()
    clc;
    debugMode = 1;
    findMax = 0;

    matr = [
        10 12 7 11 10;
        12 5 12 7 12;
        8 6 7 8 13;
        8 11 5 9 9;
        10 8 9 11 11];

    disp('15 вариант. Матрица:');
    disp(matr);
```

```

C = matr;

if findMax == 1
    C = convertToMin(matr);

    if debugMode == 1
        disp('Матрица после приведения к задаче минимизации:');
        disp(C);
    end
end

C = updateColumns(C);
if debugMode == 1
    disp('Результат вычитания наименьшего элемента по
        столбцам:');
    disp(C);
end

C = updateRows(C);
if debugMode == 1
    disp('Результат вычитания наименьшего элемента по строкам:');
    disp(C);
end

[numRows,numCols] = size(C);

matrSIZ = getSIZInit(C);
if debugMode == 1
    disp('Начальная СНН:');
    printSIZ(C, matrSIZ);
end

k = sum(sum(matrSIZ, 1));
if debugMode == 1
    fprintf('Число нулей в построенной СНН: k = %d\n\n', k);
end

iteration = 1;
while k < numCols
    if debugMode == 1

```

```

        fprintf('----- Итерация %d -----\n',
            iteration);
    end

    matrStreak = zeros(numRows, numCols);    % матрица, в которой
        отмечаются позиции 0'
    selectedColumns = sum(matrSIZ);
    selectedRows = zeros(numRows);
    selection = getSelection(numRows, numCols, selectedColumns);

    if debugMode == 1
        disp('Результат выделения столбцов, в которых стоит 0*');
        printMarkedMatr(C, matrSIZ, matrStreak, selectedColumns,
            selectedRows);
    end

    flag = true;
    streakPnt = [-1 -1];
    while flag
        if debugMode == 1
            disp('Поиск 0 среди невыделенных элементов');
        end

        streakPnt = findStreak(C, selection);
        if streakPnt(1) == -1
            C = updateMatrNoZero(C, numRows, numCols, selection,
                selectedRows, selectedColumns);

            if debugMode == 1
                disp('Т.к. среди невыделенных элементов нет нулей,
                    матрица была преобразована:');
                printMarkedMatr(C, matrSIZ, matrStreak,
                    selectedColumns, selectedRows);
            end

            streakPnt = findStreak(C, selection);
        end

        matrStreak(streakPnt(1), streakPnt(2)) = 1;
        if debugMode == 1
            disp('Матрица с найденным 0-штрих');
        end
    end

```

```

        printMarkedMatr(C, matrSIZ, matrStreak, selectedColumns,
            selectedRows);
    end

    zeroStarInRow = getZeroStarInRow(streakPnt, numCols,
        matrSIZ);
    if zeroStarInRow(1) == -1
        flag = false;
    else
        % снять выделение со столбца с 0*
        selection(:, zeroStarInRow(2)) = selection(:,
            zeroStarInRow(2)) - 1;
        selectedColumns(zeroStarInRow(2)) = 0;

        % перенести выделение на строку с 0'
        selection(zeroStarInRow(1), :) =
            selection(zeroStarInRow(1), :) + 1;
        selectedRows(zeroStarInRow(1)) = 1;
        if debugMode == 1
            disp('Т.к. в одной строке с 0-штрих есть 0*, было
                переброшено выделение:');
            printMarkedMatr(C, matrSIZ, matrStreak,
                selectedColumns, selectedRows);
        end
    end
end

if debugMode == 1
    disp('L-цепочка: ');
end

[matrStreak, matrSIZ] = createL(numRows, numCols, streakPnt,
    matrStreak, matrSIZ);

k = sum(sum(matrSIZ, 1));
if debugMode == 1
    disp('Текущая CHH:');
    printSIZ(C, matrSIZ);
    fprintf('Итого, k = %d\n', k);
end

```



```

        iteration = iteration + 1;
        disp('-----');
    end

    disp('Конечная СНН:');
    printSIZ(C, matrSIZ);

    disp('X =');
    disp(matrSIZ);

    fOpt = getFOpt(matr, matrSIZ);
    fprintf("Результат = %d\n", fOpt);
end

% Найти первый нулевой элемент среди невыделенных, в одной
% строке с которым не
% стоит 0*
function [streakPnt] = findStreak(matr, selection)
    streakPnt = [-1 -1];
    [numRows,numCols] = size(matr);
    for i = 1 : numCols
        for j = 1 : numRows
            if selection(j, i) == 0 && matr(j, i) == 0
                streakPnt(1) = j;
                streakPnt(2) = i;
                return;
            end
        end
    end
end

end

function [] = printSIZ(matr, matrSIZ)
    [numRows,numCols] = size(matr);

    fprintf("\n");
    for i = 1 : numRows
        for j = 1 : numCols
            if matrSIZ(i, j) == 1
                fprintf("\t%d*", matr(i, j));
            else

```

```

        fprintf("\t%d", matr(i, j));
    end
end
fprintf("\n");
end
fprintf("\n");
end

function [] = printMarkedMatr(matr, matrSIZ, matrStreak,
    selectedCols, selectedRows)
[numRows, numCols] = size(matr);

for i = 1 : numRows
    if selectedRows(i) == 1
        fprintf("+")
    end

    for j = 1 : numCols
        fprintf("\t%d", matr(i, j))
        if matrSIZ(i, j) == 1
            fprintf("*");
        elseif matrStreak(i, j) == 1
            fprintf("'")
        end
    end
end

    fprintf('\n');
end

for i = 1 : numCols
    if selectedCols(i) == 1
        fprintf("\t+")
    else
        fprintf(" \t")
    end
end
fprintf('\n\n');
end

% Для случая задачи максимизации - привести её к задаче
    минимизации

```

```

function matr = convertToMin(matr)
    maxElem = max(max(matr));
    matr = matr * (-1) + maxElem;
end

% В каждом столбце С нах. наим. эл-т и вычесть его из соотв.
    столбца
function matr = updateColumns(matr)
    minElemArr = min(matr);
    for i = 1 : length(minElemArr)
        matr(:, i) = matr(:, i) - minElemArr(i);
    end
end

% В каждой строке С нах. наим. эл-т и вычесть его из соотв.
    строки
function matr = updateRows(matr)
    minElemArr = min(matr, [], 2);
    for i = 1 : length(minElemArr)
        matr(i, :) = matr(i, :) - minElemArr(i);
    end
end

% Начальное состояние CHH
function matrSIZ = getSIZInit(matr)
    [numRows, numCols] = size(matr);
    matrSIZ = zeros(numRows, numCols);

    for i = 1: numCols
        for j = 1 : numRows
            if matr(j, i) == 0
                count = 0;
                for k = 1 : numCols
                    count = count + matrSIZ(j, k);
                end
                for k = 1 : numRows
                    count = count + matrSIZ(k, i);
                end
                if count == 0
                    matrSIZ(j, i) = 1;
                end
            end
        end
    end
end

```

```

        end
    end
end
end

% Выделить столбцы, в которых стоит 0*
function [selection] = getSelection(numRows, numCols,
    selectedColumns)
    selection = zeros(numRows, numCols);
    for i = 1 : numCols
        if selectedColumns(i) == 1
            selection(:, i) = selection(:, i) + 1;
        end
    end
end

% Изменить матрицу в случае, если среди невыделенных элементов
нет нуля
function [matr] = updateMatrNoZero(matr, numRows, numCols,
    selection, selectedRows, selectedColumns)
    h = 1e5; % Наименьший элемент среди невыделенных
    for i = 1 : numCols
        for j = 1 : numRows
            if selection(j, i) == 0 && matr(j, i) < h
                h = matr(j, i);
            end
        end
    end

    for i = 1 : numCols
        if selectedColumns(i) == 0
            matr(:, i) = matr(:, i) - h;
        end
    end

    for i = 1 : numRows
        if selectedRows(i) == 1
            matr(i, :) = matr(i, :) + h;
        end
    end
end
end

```

```

% Найти 0* в той же строке, что и 0'
function [zeroStarInRow] = getZeroStarInRow(streakPnt, numCols,
    matrSIZ)
    j = streakPnt(1);
    zeroStarInRow = [-1 -1];
    for i = 1 : numCols
        if matrSIZ(j, i) == 1
            zeroStarInRow(1) = j;
            zeroStarInRow(2) = i;
            break
        end
    end
end

% Построить L-цепочку
function [matrStreak, matrSIZ] = createL(numRows, numCols,
    streakPnt, matrStreak, matrSIZ)
    i = streakPnt(1);
    j = streakPnt(2);
    while i > 0 && j > 0 && i <= numRows && j <= numCols
        % Снять *
        matrStreak(i, j) = 0;

        % Заменить ' на *
        matrSIZ(i, j) = 1;

        fprintf("[%d, %d] ", i, j);

        % Дойти до 0* по столбцу от 0'
        kRow = 1;
        while kRow <= numRows && (matrSIZ(kRow, j) ~= 1 || kRow ==
            i)
            kRow = kRow + 1;
        end

        if (kRow <= numRows)
            % Дойти до 0' по столбцу от 0*
            lCol = 1;
            while lCol <= numCols && (matrStreak(kRow, lCol) ~= 1 ||
                lCol == j)
                lCol = lCol + 1;
            end
        end
    end
end

```

```

        end

        if lCol <= numCols
            matrSIZ(kRow,j) = 0;
            fprintf("-> [%d, %d] -> ", kRow, j);
        end
        j = lCol;
    end
    i = kRow;
end

    fprintf("\n");
end

function [fOpt] = getFOpt(matr, matrSIZ)
    fOpt = 0;
    [numRows,numCols] = size(matr);

    for i = 1 : numCols
        for j = 1 : numRows
            if matrSIZ(j, i) == 1
                fOpt = fOpt + matr(j, i);
            end
        end
    end
end
end

```

## 2.3 Примеры работы программы

Листинг 2.2 – Решение задачи минимизации

15 вариант. Матрица:

|    |    |    |    |    |
|----|----|----|----|----|
| 10 | 12 | 7  | 11 | 10 |
| 12 | 5  | 12 | 7  | 12 |
| 8  | 6  | 7  | 8  | 13 |
| 8  | 11 | 5  | 9  | 9  |
| 10 | 8  | 9  | 11 | 11 |

Результат вычитания наименьшего элемента по столбцам:

|   |   |   |   |   |
|---|---|---|---|---|
| 2 | 7 | 2 | 4 | 1 |
| 4 | 0 | 7 | 0 | 3 |
| 0 | 1 | 2 | 1 | 4 |
| 0 | 6 | 0 | 2 | 0 |

2 3 4 4 2

Результат вычитания наименьшего элемента по строкам:

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 6 | 1 | 3 | 0 |
| 4 | 0 | 7 | 0 | 3 |
| 0 | 1 | 2 | 1 | 4 |
| 0 | 6 | 0 | 2 | 0 |
| 0 | 1 | 2 | 2 | 0 |

Начальная СНН:

|    |    |    |   |    |
|----|----|----|---|----|
| 1  | 6  | 1  | 3 | 0* |
| 4  | 0* | 7  | 0 | 3  |
| 0* | 1  | 2  | 1 | 4  |
| 0  | 6  | 0* | 2 | 0  |
| 0  | 1  | 2  | 2 | 0  |

Число нулей в построенной СНН:  $k = 4$

----- Итерация 1 -----

Результат выделения столбцов, в которых стоит 0\*:

|    |    |    |   |    |
|----|----|----|---|----|
| 1  | 6  | 1  | 3 | 0* |
| 4  | 0* | 7  | 0 | 3  |
| 0* | 1  | 2  | 1 | 4  |
| 0  | 6  | 0* | 2 | 0  |
| 0  | 1  | 2  | 2 | 0  |
| +  | +  | +  |   | +  |

Поиск 0 среди невыделенных элементов

Матрица с найденным 0-штрих

|    |    |    |    |    |
|----|----|----|----|----|
| 1  | 6  | 1  | 3  | 0* |
| 4  | 0* | 7  | 0' | 3  |
| 0* | 1  | 2  | 1  | 4  |
| 0  | 6  | 0* | 2  | 0  |
| 0  | 1  | 2  | 2  | 0  |
| +  | +  | +  |    | +  |

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

|   |    |    |    |    |    |
|---|----|----|----|----|----|
|   | 1  | 6  | 1  | 3  | 0* |
| + | 4  | 0* | 7  | 0' | 3  |
|   | 0* | 1  | 2  | 1  | 4  |
|   | 0  | 6  | 0* | 2  | 0  |

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 1 | 2 | 2 | 0 |
| + |   | + |   | + |

Поиск 0 среди невыделенных элементов

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

|   |    |    |    |    |    |
|---|----|----|----|----|----|
|   | 1  | 5  | 1  | 2  | 0* |
| + | 5  | 0* | 8  | 0' | 4  |
|   | 0* | 0  | 2  | 0  | 4  |
|   | 0  | 5  | 0* | 1  | 0  |
|   | 0  | 0  | 2  | 1  | 0  |
|   | +  |    | +  |    | +  |

Матрица с найденным 0-штрих

|   |    |    |    |    |    |
|---|----|----|----|----|----|
|   | 1  | 5  | 1  | 2  | 0* |
| + | 5  | 0* | 8  | 0' | 4  |
|   | 0* | 0' | 2  | 0  | 4  |
|   | 0  | 5  | 0* | 1  | 0  |
|   | 0  | 0  | 2  | 1  | 0  |
|   | +  |    | +  |    | +  |

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

|   |    |    |    |    |    |
|---|----|----|----|----|----|
|   | 1  | 5  | 1  | 2  | 0* |
| + | 5  | 0* | 8  | 0' | 4  |
| + | 0* | 0' | 2  | 0  | 4  |
|   | 0  | 5  | 0* | 1  | 0  |
|   | 0  | 0  | 2  | 1  | 0  |
|   |    |    | +  |    | +  |

Поиск 0 среди невыделенных элементов

Матрица с найденным 0-штрих

|   |    |    |    |    |    |
|---|----|----|----|----|----|
|   | 1  | 5  | 1  | 2  | 0* |
| + | 5  | 0* | 8  | 0' | 4  |
| + | 0* | 0' | 2  | 0  | 4  |
|   | 0' | 5  | 0* | 1  | 0  |
|   | 0  | 0  | 2  | 1  | 0  |
|   |    |    | +  |    | +  |

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



|   |    |    |    |    |    |
|---|----|----|----|----|----|
|   | 1  | 5  | 1  | 2  | 0* |
| + | 5  | 0* | 8  | 0' | 4  |
| + | 0* | 0' | 2  | 0  | 4  |
| + | 0' | 5  | 0* | 1  | 0  |
|   | 0  | 0  | 2  | 1  | 0  |
|   |    |    |    |    | +  |

Поиск 0 среди невыделенных элементов

Матрица с найденным 0-штрих

|   |    |    |    |    |    |
|---|----|----|----|----|----|
|   | 1  | 5  | 1  | 2  | 0* |
| + | 5  | 0* | 8  | 0' | 4  |
| + | 0* | 0' | 2  | 0  | 4  |
| + | 0' | 5  | 0* | 1  | 0  |
|   | 0' | 0  | 2  | 1  | 0  |
|   |    |    |    |    | +  |

L-цепочка:

[5, 1] -> [3, 1] -> [3, 2] -> [2, 2] -> [2, 4]

Текущая СНН:

|  |    |    |    |    |    |
|--|----|----|----|----|----|
|  | 1  | 5  | 1  | 2  | 0* |
|  | 5  | 0  | 8  | 0* | 4  |
|  | 0  | 0* | 2  | 0  | 4  |
|  | 0  | 5  | 0* | 1  | 0  |
|  | 0* | 0  | 2  | 1  | 0  |

Итого, k = 5

-----  
Конечная СНН:

|  |    |    |    |    |    |
|--|----|----|----|----|----|
|  | 1  | 5  | 1  | 2  | 0* |
|  | 5  | 0  | 8  | 0* | 4  |
|  | 0  | 0* | 2  | 0  | 4  |
|  | 0  | 5  | 0* | 1  | 0  |
|  | 0* | 0  | 2  | 1  | 0  |

X =

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|

Результат = 38

### Листинг 2.3 – Решение задачи максимизации

15 вариант. Матрица:

|    |    |    |    |    |
|----|----|----|----|----|
| 10 | 12 | 7  | 11 | 10 |
| 12 | 5  | 12 | 7  | 12 |
| 8  | 6  | 7  | 8  | 13 |
| 8  | 11 | 5  | 9  | 9  |
| 10 | 8  | 9  | 11 | 11 |

Матрица после приведения к задаче минимизации:

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

Результат вычитания наименьшего элемента по столбцам:

|   |   |   |   |   |
|---|---|---|---|---|
| 2 | 0 | 5 | 0 | 3 |
| 0 | 7 | 0 | 4 | 1 |
| 4 | 6 | 5 | 3 | 0 |
| 4 | 1 | 7 | 2 | 4 |
| 2 | 4 | 3 | 0 | 2 |

Результат вычитания наименьшего элемента по строкам:

|   |   |   |   |   |
|---|---|---|---|---|
| 2 | 0 | 5 | 0 | 3 |
| 0 | 7 | 0 | 4 | 1 |
| 4 | 6 | 5 | 3 | 0 |
| 3 | 0 | 6 | 1 | 3 |
| 2 | 4 | 3 | 0 | 2 |

Начальная СНН:

|    |    |   |    |    |
|----|----|---|----|----|
| 2  | 0* | 5 | 0  | 3  |
| 0* | 7  | 0 | 4  | 1  |
| 4  | 6  | 5 | 3  | 0* |
| 3  | 0  | 6 | 1  | 3  |
| 2  | 4  | 3 | 0* | 2  |

Число нулей в построенной СНН:  $k = 4$

----- Итерация 1 -----

Результат выделения столбцов, в которых стоит 0\*:

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

|   |   |   |    |    |
|---|---|---|----|----|
| 4 | 6 | 5 | 3  | 0* |
| 3 | 0 | 6 | 1  | 3  |
| 2 | 4 | 3 | 0* | 2  |
| + | + |   | +  | +  |

Поиск 0 среди невыделенных элементов

Матрица с найденным 0-штрих

|    |    |    |    |    |
|----|----|----|----|----|
| 2  | 0* | 5  | 0  | 3  |
| 0* | 7  | 0' | 4  | 1  |
| 4  | 6  | 5  | 3  | 0* |
| 3  | 0  | 6  | 1  | 3  |
| 2  | 4  | 3  | 0* | 2  |
| +  | +  |    | +  | +  |

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

|   |    |    |    |    |    |
|---|----|----|----|----|----|
|   | 2  | 0* | 5  | 0  | 3  |
| + | 0* | 7  | 0' | 4  | 1  |
|   | 4  | 6  | 5  | 3  | 0* |
|   | 3  | 0  | 6  | 1  | 3  |
|   | 2  | 4  | 3  | 0* | 2  |
|   |    | +  |    | +  | +  |

Поиск 0 среди невыделенных элементов

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

|   |    |    |    |    |    |
|---|----|----|----|----|----|
|   | 0  | 0* | 3  | 0  | 3  |
| + | 0* | 9  | 0' | 6  | 3  |
|   | 2  | 6  | 3  | 3  | 0* |
|   | 1  | 0  | 4  | 1  | 3  |
|   | 0  | 4  | 1  | 0* | 2  |
|   |    | +  |    | +  | +  |

Матрица с найденным 0-штрих

|   |    |    |    |    |    |
|---|----|----|----|----|----|
|   | 0' | 0* | 3  | 0  | 3  |
| + | 0* | 9  | 0' | 6  | 3  |
|   | 2  | 6  | 3  | 3  | 0* |
|   | 1  | 0  | 4  | 1  | 3  |
|   | 0  | 4  | 1  | 0* | 2  |
|   |    | +  |    | +  | +  |

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

|   |    |    |    |    |    |
|---|----|----|----|----|----|
| + | 0' | 0* | 3  | 0  | 3  |
| + | 0* | 9  | 0' | 6  | 3  |
|   | 2  | 6  | 3  | 3  | 0* |
|   | 1  | 0  | 4  | 1  | 3  |
|   | 0  | 4  | 1  | 0* | 2  |
|   |    |    |    | +  | +  |

Поиск 0 среди невыделенных элементов

Матрица с найденным 0-штрих

|   |    |    |    |    |    |
|---|----|----|----|----|----|
| + | 0' | 0* | 3  | 0  | 3  |
| + | 0* | 9  | 0' | 6  | 3  |
|   | 2  | 6  | 3  | 3  | 0* |
|   | 1  | 0  | 4  | 1  | 3  |
|   | 0' | 4  | 1  | 0* | 2  |
|   |    |    |    | +  | +  |

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

|   |    |    |    |    |    |
|---|----|----|----|----|----|
| + | 0' | 0* | 3  | 0  | 3  |
| + | 0* | 9  | 0' | 6  | 3  |
|   | 2  | 6  | 3  | 3  | 0* |
|   | 1  | 0  | 4  | 1  | 3  |
| + | 0' | 4  | 1  | 0* | 2  |
|   |    |    |    |    | +  |

Поиск 0 среди невыделенных элементов

Матрица с найденным 0-штрих

|   |    |    |    |    |    |
|---|----|----|----|----|----|
| + | 0' | 0* | 3  | 0  | 3  |
| + | 0* | 9  | 0' | 6  | 3  |
|   | 2  | 6  | 3  | 3  | 0* |
|   | 1  | 0' | 4  | 1  | 3  |
| + | 0' | 4  | 1  | 0* | 2  |
|   |    |    |    |    | +  |

L-цепочка:

[4, 2] -> [1, 2] -> [1, 1] -> [2, 1] -> [2, 3]

Текущая СНН:

|    |   |   |   |   |
|----|---|---|---|---|
| 0* | 0 | 3 | 0 | 3 |
|----|---|---|---|---|

|   |    |    |    |    |
|---|----|----|----|----|
| 0 | 9  | 0* | 6  | 3  |
| 2 | 6  | 3  | 3  | 0* |
| 1 | 0* | 4  | 1  | 3  |
| 0 | 4  | 1  | 0* | 2  |

Итого,  $k = 5$

-----

Конечная СНН:

|    |    |    |    |    |
|----|----|----|----|----|
| 0* | 0  | 3  | 0  | 3  |
| 0  | 9  | 0* | 6  | 3  |
| 2  | 6  | 3  | 3  | 0* |
| 1  | 0* | 4  | 1  | 3  |
| 0  | 4  | 1  | 0* | 2  |

$X =$

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |

Результат = 57