Министерство образования и науки Российской Федерации Федеральное агентство по образованию Федеральное государственное бюджетное образовательное учреждение высшего образования «Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Лабораторная работа №4

по курсу «Исследование операций»

**Транспортная задача**

Выполнил студент группы ИВТ-31\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Птахова А.М/

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Коржавина А.С./

Киров 2022

1. Цель

Закрепить на практике знания о методах решения транспортных задач линейного программирования и получить навыки их программной реализации.

2. Задание

- выбрать метод первоначального заполнения базиса и метод решения транспортной задачи, согласовать выбор задач с преподавателем.

- реализовать выбранные методы решения транспортной задачи

3. Описание метода

3.1. Венгерский метод

На предварительном этапе определяется начальный план, и рассчитываются невязки этого плана по строкам, столбцам и суммарная невязка.

Если при этом окажется, что суммарная невязка нулевая, то это означает оптимальность полученного плана. В противном случае, план необходимо улучшать, для чего и выполняется итерационная часть алгоритма.

Итерация, в свою очередь, состоит из нескольких этапов, которым предшествует разметка:

- поиск невыделенных нулей в строках с положительными невязками.

- коррекция плана X и невязок путем построения цепочки.

- эквивалентные преобразования матрицы C. Итерация начинается первым этапом, а заканчивается вторым, между которыми может производиться несколько третьих.

3.2. Северо-западный угол

Формирование опорного плана начинаем с внесения в верхнюю левую клетку максимально возможного объема перевозки.

Запасы на складе A1 закончились, поэтому в оставшиеся ячейки данной строки ставим прочерки. Затем переходим к следующей строке и заполняем ее ячейки слева направо.

Переходим к третьей строке и тоже заполняем ее слева направо.

4.Листинг кода

#include <iostream>

#include <vector>

using namespace std;

void input(vector <vector <double>>& mas, int n, int m)

{

/\*for (int i = 0; i < n; i++)

{

for (int j = 0; j < m; j++)

{

cout << "mas[" << i << "," << j << "]= ";

cin >> mas[i][j];

}

}\*/

mas[0][0] = 3; mas[0][1] = 4; mas[0][2] = 6; mas[0][3] = 4; mas[0][4] = 10; mas[0][5] = 140;

mas[1][0] = 9; mas[1][1] = 10; mas[1][2] = 15; mas[1][3] = 12; mas[1][4] = 6; mas[1][5] = 50;

mas[2][0] = 8; mas[2][1] = 8; mas[2][2] = 14; mas[2][3] = 8; mas[2][4] = 5; mas[2][5] = 260;

mas[3][0] = 150; mas[3][1] = 170; mas[3][2] = 30; mas[3][3] = 90; mas[3][4] = 10; mas[3][5] = 450;

}

void output(vector <vector <double>>& mas, int n, int m)

{

for (int i = 0; i < n; i++)

{

for (int j = 0; j < m; j++)

{

cout << mas[i][j]<<' ';

}

cout << '\n';

}

}

void min\_str(vector <vector <double>>& mas, vector <double> &min, int n, int m)

{

for (int i = 0; i < n-1; i++)

{

for (int j = 0; j < m-1; j++)

{

if (mas[i][j] < min[i])

{

min[i] = mas[i][j];

}

}

}

}

void min\_row(vector <vector <double>>& mas, vector <double>& min, int n, int m)

{

for (int j = 0; j < m - 1; j++)

{

for (int i = 0; i < n - 1; i++)

{

/\*cout << mas[i][j] << ' ';\*/

if (mas[i][j] < min[j])

{

min[j] = mas[i][j];

}

}

//cout << '\n';

}

}

void sub\_str(vector <vector <double>>& mas, vector <double>& min, int n, int m)

{

for (int i = 0; i < n ; i++)

{

for (int j = 0; j < m; j++)

{

if (min[i] != 1000)

{

mas[i][j] = mas[i][j] - min[i];

}

}

}

}

void sub\_row(vector <vector <double>>& mas, vector <double>& min, int n, int m)

{

for (int j = 0; j < m; j++)

{

for (int i = 0; i < n; i++)

{

if (min[i] != 1000)

{

mas[i][j] = mas[i][j] - min[j];

}

}

}

}

void modifacation(vector <vector <double>>& mas, int n, int m)

{

int min = 1000;

for (int j = 0; j < m; j++)

{

if (mas[1][j] < min && mas[1][j]!=0)

{

min = mas[1][j];

}

}

for (int j = 0; j < m; j++)

{

mas[1][j] = mas[1][j] - min;

}

for (int i = 0; i < n; i++)

{

mas[i][m-1] = mas[i][m-1] + min;

}

}

void modifacation2(vector <vector <double>>& mas, int n, int m)

{

int min = 1000;

for (int j = 0; j < m; j++)

{

if (mas[2][j] < min && mas[2][j] != 0)

{

min = mas[2][j];

}

}

for (int j = 0; j < m; j++)

{

mas[2][j] = mas[2][j] - min;

}

}

int check(vector <vector <double>>& mas, int n, int m)

{

int k = 0;

int ko = 0;

for (int j = 0; j < m; j++)

{

for (int i = 0; i < n; i++)

{

if (mas[i][j] == 0)

{

k++;

}

}

if (k < 2)

{

ko++;

k = 0;

}

}

return ko;

}

void perenos(vector <vector <double>>& mas, vector <vector <double>>& mas1, int n, int m)

{

for (int i = 0; i < n; i++)

{

mas1[i][m-1] = mas[i][m-1];

}

for (int j = 0; j < m; j++)

{

mas1[n-1][j] = mas[n-1][j];

}

}

void north\_west(vector <vector <double>>& mas, vector <vector <double>>& mas1, int n, int m)

{

int z = 0;

for (int i = n - 1; i >= 0; i--)

{

for (int j = m - 1; j >= 0; j--)

{

if (mas[i][j] == 0)

{

z = min(mas[i][m], mas[n][j]);

mas1[i][j] = z;

mas[i][m] = mas[i][m] - z;

mas[n][j] = mas[n][j] - z;

//cout << z <<'\n';

//cout << mas1[i][j] << ' ';

}

}

}

}

double result(vector <vector <double>>& copy, vector <vector <double>>& mas1, int n, int m)

{

double z = 0;

for (int i = 0; i < n-1; i++)

{

for (int j = 0; j < m-1; j++)

{

//cout << copy[i][j]<<' ';

z = z + mas1[i][j] \* copy[i][j];

}

}

return z;

}

int main()

{

setlocale(LC\_ALL, "rus");

int n;

int m;

cout << "Введите кол-во источников: ";

cin >> n; //string

cout << "Введите кол-во пунктов назначения: ";

cin >> m; //row

int o;

int d;

d = n + 1;

o = m + 1;

vector <vector <double>> mas(d, vector <double>(o));//inizialization

vector <vector <double>> copy(d, vector <double>(o));//inizialization

input(mas, d, o);//inizialization

output(mas, d, o);

cout << '\n';

copy = mas;

//output(copy, d, o); //checking

//cout << '\n';

vector <double> min(m, 1000);

#ifdef DEBUG

for (int i = 0; i < m; i++) //for debug

{

cout << min[i] << ' ';

}

#endif // DEBUG

min\_str(mas, min, d, o); // find min in everyone string

#ifdef DEBUG

for (int i = 0; i < m; i++) //for debug

{

cout << min[i] << ' ';

}

#endif // DEBUG

sub\_str(mas, min, n, m); //sub min and element of string

output(mas, d, o); //checking

cout << '\n';

#ifdef DEBUG

for (int i = 0; i < m; i++) //for debug

{

min[i] =1000;

}

#endif // DEBUG

min\_row(mas, min, d, o);//find min in the row

#ifdef DEBUG

for (int i = 0; i < m; i++) //for debug

{

cout << min[i] << ' ';

}

#endif // DEBUG

sub\_row(mas, min, n, m); //sub min and element of row

output(mas, d, o); //checking

cout << '\n';

int count = check(mas, n, m);

cout <<"Количеcтво: " << count << '\n';

cout << '\n';

vector <vector <double>> mas1(d, vector <double>(o));//inizialization

perenos(mas, mas1, d, o);

if (count > n)

{

cout << "Требуется модификация"<<'\n';

modifacation(mas, n, m);

modifacation2(mas, n, m);

output(mas, d, o); //checking

cout << '\n';

north\_west(mas,mas1, n, m);

output(mas1, d, o); //checking

cout << '\n';

double res = result(copy, mas1, d, o);

cout <<"Ответ: " << res << '\n';

}

else

{

cout << "Не требуются модификация" << '\n';

north\_west(mas, mas1, n, m);

output(mas1, d, o); //checking

cout << '\n';

double res = result(copy, mas1, d, o);

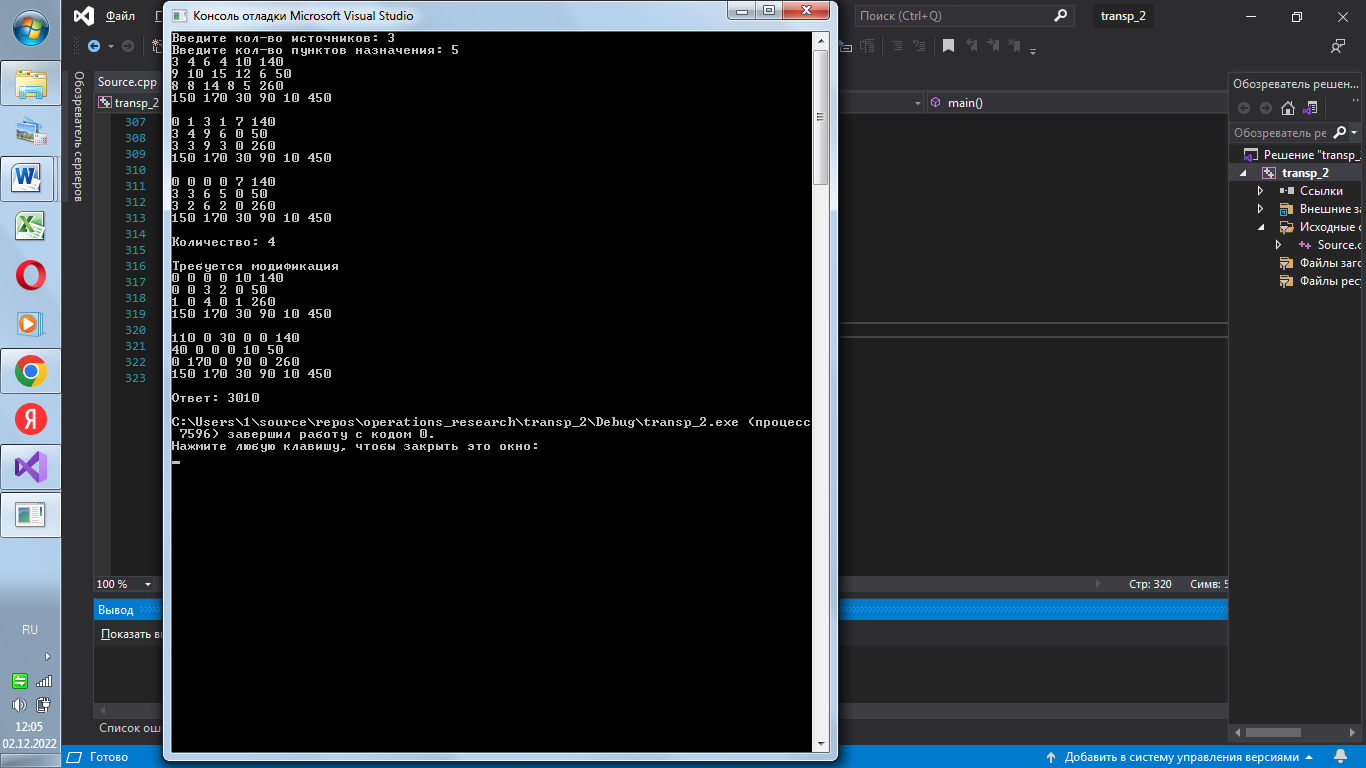
cout << "Ответ: " << res << '\n';

}

return 0;

}

5. Экранные формы



6. Выводы

В ходе выполнения лабораторной работы были получены знания о методах решения транспортной задачи и навыки реализации венгерского метода и метода северо-западного угла на языке высокого уровня.