Федеральное государственное бюджетное образовательное учреждение «Сибирский государственный университет телекоммуникаций и информатики»

Кафедра ПМиК

Лабораторная работа №3

по дисциплине

«Алгоритмы и вычислительные методы оптимизации»

Выполнил: студент III курса

Группы ИП-813

Бурдуковский И.А.

Проверила:

Галкина М.Ю.

Новосибирск 2021

Оглавление

[Задание на лабораторную работу 3](#_Toc40795258)

[Текст программы 4](#_Toc40795259)

[Результаты тестирования 12](#_Toc40795260)

# Задание на лабораторную работу

Нахождение начального опорного плана транспортной задачи

Написать программу, находящую начальный опорный план транспортной задачи методом минимальной стоимости.

Матрицу тарифов, запасы поставщиков и потребности потребителей вводить из файла. Программа должна работать как с открытой, так и закрытой моделью транспортной задачи. Предусмотреть программное нахождение вырожденного плана. Вывести распределение перевозок и затраты. Для тестирования использовать несколько заданий из практических занятий.

# Текст программы

#include <fstream>

#include <conio.h>

#include <iomanip>

#include <cstdio>

#include <cstring>

#include <cstdlib>

#include <iostream>

#include <sstream>

using namespace std;

int n,m; //n- columns, m - lines

//структура минимумов для строчек / столбцов

struct minimum{

int min\_first = 9999999;

int min\_second = 9999999;

int i = -1;

int j = -1;

int dif = 0;

};

//обнуление массива структур разниц минимумов на новой итерации

void set\_struct\_to\_default(minimum \*mas, int size){

for(int i = 0; i < size; i++){

mas[i].min\_first = 9999999;

mas[i].min\_second = 9999999;

mas[i].i = -1;

mas[i].j = -1;

mas[i].dif = 0;

}

return;

}

void print(int \*\*mas){

cout << endl << " Поставщики";

for(int i = 1; i < n; i++){

cout << "\t B" << i;

}

cout << "\tЗапасы" << endl;

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

{

if(i == m-1){

cout << "Потребности";

}

else{

cout << "\tA" << i+1;

}

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

{

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

printf("\t%3d",mas[i][j]);

}

else{

if(mas[i][j] == -10){

printf("\t -");

}

}

}

cout << endl;

}

cout << endl;

}

void print\_with\_min(int \*\*mas, minimum \*min\_lines, minimum \*min\_colls){

cout << endl << " Поставщики";

for(int i = 1; i < n; i++){

cout << "\t B" << i;

}

cout << "\tЗапасы\tМин" << endl;

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

{

if(i == m-1){

cout << "Потребности";

}

else{

cout << "\tA" << i+1;

}

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

{

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

printf("\t%3d",mas[i][j]);

}

else{

if(mas[i][j] == -10){

printf("\t -");

}

}

}

if(i < m-1){

if(min\_lines[i].dif == -10){

printf("\t -\n");

}

else{

printf("\t%3d\n",min\_lines[i].dif);

}

}

}

cout << endl << " Мин";

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

if(min\_colls[i].dif == -10){

printf("\t -");

}

else{

printf("\t%3d",min\_colls[i].dif);

}

}

}

int \*\*update\_Table(int \*\*arr){

int column\_sum = 0;

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

column\_sum += arr[i][n-1];

}

int line\_sum = 0;

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

line\_sum += arr[m-1][i];

}

cout << endl << "\t Запасы: " << column\_sum << "\t\t\t Потребности: " << line\_sum << endl << endl;

//проверяем сходятся ли запасы с потребностями

if(column\_sum == line\_sum){

cout << "\tЗапасы удовлетворяют потребностям, таблица не меняется\n";

return arr;

}

int \*\*new\_arr;

//если потребностей больше - добавляем ещё одну строку

if(line\_sum > column\_sum){

m = m+1;

new\_arr = new int\*[m];

for (int i=0;i<m-2;i++)

{

new\_arr[i]=new int[n];

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

{

new\_arr[i][j] = arr[i][j];

}

}

//доп строка

new\_arr[m-2] = new int[n];

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

{

new\_arr[m-2][i] = 0;

}

new\_arr[m-2][n-1] = line\_sum - column\_sum;

//последняя строка

new\_arr[m-1]=new int[n];

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

{

new\_arr[m-1][i] = arr[m-2][i];

}

cout << "\tПотребностей больше чем запасов, добавлен новый поставщик\n";

return new\_arr;

}

//если запасов больше - добавляем ещё один столбец

else{

n = n+1;

new\_arr = new int\*[m];

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

{

new\_arr[i]=new int[n];

for (int j = 0; j < n-2; j++)

{

new\_arr[i][j] = arr[i][j];

}

}

//доп столбец

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

{

new\_arr[i][n-2] = 0;

}

new\_arr[m-1][n-2] = column\_sum - line\_sum;

//последний столбец

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

{

new\_arr[i][n-1] = arr[i][n-2];

}

cout << "\tЗапасов больше чем потребностей, добавлен новый потребитель\n";

return new\_arr;

}

}

//считает разности двух минимумов для строчек

void count\_lines\_min(int \*\*arr, minimum \*mas){

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

//находим первый минимум в строке

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

if((arr[i][j] > 0) && (arr[i][j] < mas[i].min\_first)){

mas[i].min\_first = arr[i][j];

mas[i].i = i;

mas[i].j = j;

}

}

//находим второй минимум в строке

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

if((arr[i][j] > 0) && (arr[i][j] < mas[i].min\_second) && (j != mas[i].j)){

mas[i].min\_second = arr[i][j];

}

}

//находим разницу между двумя миимумами

mas[i].dif = mas[i].min\_second - mas[i].min\_first;

//если строка добавочная

if(mas[i].min\_first == 9999999){

int counter = 0;

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

if(arr[i][j] == 0){

mas[i].i = i;

mas[i].j = j;

break;

}

counter++;

}

//если спрос удовлетворён и вся строка уже закрашена

if(counter == n-1){

mas[i].dif = -10;

}

}

//если остался единственный элемент в строке

else{

if(mas[i].min\_second == 9999999){

mas[i].dif = 0;

}

}

}

/\*

cout << endl;

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

cout << mas[i].dif << " ";

}

cout << endl;

\*/

return;

}

//считает разности двух минимумов для столбцов

void count\_columns\_min(int \*\*arr, minimum \*mas){

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

//находим первый минимум в столбце

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

if((arr[i][j] > 0) && (arr[i][j] < mas[j].min\_first)){

mas[j].min\_first = arr[i][j];

mas[j].i = i;

mas[j].j = j;

}

}

//находим второй минимум в столбце

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

if((arr[i][j] > 0) && (arr[i][j] < mas[j].min\_second) && (i != mas[j].i)){

mas[j].min\_second = arr[i][j];

}

}

//находим разницу между двумя миимумами

mas[j].dif = mas[j].min\_second - mas[j].min\_first;

//если столбец добавочный

if(mas[j].min\_first == 9999999){

int counter = 0;

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

if(arr[i][j] == 0){

mas[j].i = i;

mas[j].j = j;

break;

}

counter++;

}

//если поставки исчерпаны и весь столбец закрашен

if(counter == m-1){

mas[j].dif = -10;

}

}

//если остался единственный элемент в столбце

else{

if(mas[j].min\_second == 9999999){

mas[j].dif = 0;

}

}

}

/\*

cout << endl;

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

//cout << mas[i].min\_first << "-" << mas[i].min\_second << "\n";

cout << mas[i].dif << " ";

}

cout << endl;

\*/

return;

}

minimum find\_max(minimum \*mas, int size){

minimum max = mas[0];

for(int i = 1; i < size; i++){

if(mas[i].dif >= max.dif){

max = mas[i];

}

}

return max;

}

int check\_end(int \*\*arr){

int counter = 0;

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

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

if(arr[i][j] != -10){

counter++;

}

}

}

if(counter == 0){

return 1;

}

return 0;

}

int take\_StockNeedMin(int a, int b){

if (a >= b){

return b;

}

else{

return a;

}

}

void zero\_line(int \*\*arr, int line){

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

arr[line][i] = -10;

}

}

void zero\_collumn(int \*\*arr, int line){

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

arr[i][line] = -10;

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

int \*\*arr, \*\*pre\_arr, \*\*extra\_arr;

string filename;

cout << "Введите название файла:" << endl;

cin >> filename;

ifstream input(filename);

if(!input){

cout << "Ошибка открытия файла" << endl;

return 0;

}

input >> m;

input >> n;

pre\_arr = new int\*[m];

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

{

pre\_arr[i]=new int[n];

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

{

if(i == m-1 && j == n-1){

pre\_arr[i][j] = -100;

}

else

input >> pre\_arr[i][j];

}

}

cout << "\tИсходная таблица:";

print(pre\_arr);

arr = update\_Table(pre\_arr);

cout << "\n\tНовая таблица:";

print(arr);

extra\_arr = new int\*[m];

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

{

extra\_arr[i]=new int[n];

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

{

if(j < n-1 && i < m-1){

extra\_arr[i][j] = -10;

}

else{

extra\_arr[i][j] = arr[i][j];

}

}

}

minimum min\_lines[m-1];

minimum min\_colls[n-1];

minimum result;

int Z = 0;

while(true){

//зануляем массмвы структуп минимальных элементов

set\_struct\_to\_default(min\_lines,m-1);

set\_struct\_to\_default(min\_colls,n-1);

//считаем разницы минимальных элементов по строкам/столбцам

count\_lines\_min(arr,min\_lines);

count\_columns\_min(arr,min\_colls);

print\_with\_min(arr,min\_lines,min\_colls);

//находим максимальные разницы из двух массивов

minimum line\_result = find\_max(min\_lines,m-1);

minimum coll\_result = find\_max(min\_colls,n-1);

//cout << endl << line\_result.dif;

//cout << endl << coll\_result.dif;

//определяем какую строку/столбец рассматриваем в завис-ти от максимальной разницы

if(line\_result.dif >= coll\_result.dif){

result = line\_result;

}

else{

result = coll\_result;

}

cout << endl << endl << endl << "\tМинимальный элемент: А" << result.i + 1 << "B" << result.j + 1;

//выбираем запамы или потребности

int Available\_capacity = take\_StockNeedMin(arr[result.i][n-1],arr[m-1][result.j]);

Z += arr[result.i][result.j] \* Available\_capacity;

//вычитаем возможный объём из запасов

arr[result.i][n-1] -= Available\_capacity;

extra\_arr[result.i][n-1] -= Available\_capacity;

//если запасы кончились, то закрашиваем строку

if(arr[result.i][n-1] == 0){

zero\_line(arr, result.i);

}

//вычитаем возможный объём из потребностей

arr[m-1][result.j] -= Available\_capacity;

extra\_arr[m-1][result.j] -= Available\_capacity;

//если потребности кончились, то закрашиваем столбец

if(arr[m-1][result.j] == 0){

zero\_collumn(arr, result.j);

}

arr[result.i][result.j] = -10;

extra\_arr[result.i][result.j] = Available\_capacity;

if(check\_end(arr) == 1){

break;

}

// ставит дополнительный 0, фиг знает зачем оно если нигде не используется

if((arr[result.i][n-1] == 0) && (arr[m-1][result.j] == 0)){

if((result.i - 1 > 0) && (extra\_arr[result.i-1][result.j] == -10)){

extra\_arr[result.i-1][result.j] = 0;

}

else{

if((result.i + 1 > 0) && (extra\_arr[result.i + 1][result.j] == -10)){

extra\_arr[result.i+1][result.j] = 0;

}

else{

if((result.j - 1 > 0) && (extra\_arr[result.i][result.j-1] == -10)){

extra\_arr[result.i][result.j-1] = 0;

}

else{

if((result.j + 1 > 0) && (extra\_arr[result.i][result.j+1] == -10)){

extra\_arr[result.i][result.j+1] = 0;

}

}

}

}

}

cout << endl;

print(extra\_arr);

}

cout << endl;

print(extra\_arr);

print(arr);

cout << "\t Z = " << Z;

return 0;

}

# Результаты тестирования





 