Министерство образования и науки РФ

Федеральное государственное автономное образовательное учреждение высшего образования

«Омский государственный технический университет»

|  |  |
| --- | --- |
| Факультет (институт) | *Информационных технологий и компьютерных систем* |
|  |  |
| Кафедра | *Прикладная математика и фундаментальная информатика* |
|  |  |

**Расчетно-графическая работа**

|  |  |
| --- | --- |
| по дисциплине | ***Дискретная математика*** |
|  |  |

Пояснительная записка

|  |  |
| --- | --- |
| **Шифр проекта** | 021-РГР-02.03.02-№ 6-ПЗ |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Студента** | | Сагалбаева Дамира Амангельдыевича | | | | | |
|  |  |  |  | | фамилия, имя, отчество полностью | | | | | |
|  |  |  | Курс | *1* |  | Группа | | ФИТ-**222** | | |
|  |  |  |  |  |  | |  |  | |  |
|  | | | **Направление (специальность)** | | | | | ***02.03.02*** | | |
|  | | | *Фундаментальная информатика и информационные технологии* | | | | | | | |
|  |  |  | код, наименование | | | | | | | |
|  |  |  | Руководитель | | ***ст. преподаватель*** | | | | | |
|  |  |  | ученая степень, звание | | | | | |
|  |  |  | ***Федотова И.В.*** | | | | | | | |
|  |  |  | фамилия, инициалы | | | | | | | |
|  |  |  | Выполнил | |  | | | | | |
|  |  |  | дата, подпись студента | | | | | |
|  |  |  | **Работа защищена с количеством баллов** | | | | | | | |
|  |  |  |  | | | | | |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | дата, подпись руководителя |  |  |  |

**Омск 2023**

**Содержание**

Задание 1 3

Текст программы на языке С# 4

Файл Program.cs 4

Задание 2 5

Текст программы на языке С# 6

Файл Program.cs 6

Файл input.txt 8

Файл output.txt………………………………………………………………………..9

# **Задание 1**

# Реализовать алгоритм вычисления определителя матрицы.

# **Текст программы на языке С#**

**Текст файла Program.cs:**

using System.ComponentModel;

static double Determinant(double[,] matrix)

{

int rows = matrix.GetLength(0);

int columns = matrix.GetLength(1);

if (rows != columns)

throw new ArgumentException("Матрица должна быть квадратной");

double det = 0;

if (rows == 2)

{

det = matrix[0, 0] \* matrix[1, 1] - matrix[0, 1] \* matrix[1, 0];

}

else

{

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

{

double[,] submatrix = new double[rows - 1, columns - 1];

for (int i = 1; i < rows; i++)

{

for (int k = 0; k < columns; k++)

{

if (k < j)

submatrix[i - 1, k] = matrix[i, k];

else if (k > j)

submatrix[i - 1, k - 1] = matrix[i, k];

}

}

det += Math.Pow(-1, j) \* matrix[0, j] \* Determinant(submatrix);

}

}

return det;

}

Console.WriteLine("Введите размерность матрицы");

int n=Int32.Parse(Console.ReadLine());

double[,] mat = new double[n,n];

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

{

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

{

mat[i, j] = Convert.ToDouble(Console.ReadLine());

}

}

Console.WriteLine(Determinant(mat));

# **Задание 2**

С целью подготовки к проведению чемпионата по информатики мэр решил обеспечить надежным электроснабжением все школы города. Для этого необходимо соединить линиями электропередач школы между собой.

Известна стоимость соединения между некоторыми парами школ. Мэр города решил выбрать одну из двух наиболее экономичных схем электроснабжения (стоимость схемы равняется сумме стоимостей соединений пар школ).

Напишите программу, которая вычисляет стоимость двух наиболее экономных схем альтернативного электроснабжения школ.

**Формат входных данных**

В первой строке входного файла два натуральных числа N и M – количество школ в городе и количество возможных соединений между ними соответственно (3 ≤ N ≤ 100, N ≤ M ≤ N∙(N-1)/2).

В каждой из последующих M строк находятся по три целых числа: Ai, Bi, Ci, где Ci – стоимость прокладки линии электроснабжения от школы Ai до школы Bi (1 ≤ Ci ≤ 300; i=1, 2, … , N). Гарантируется, что существует хотя бы две различные схемы электроснабжения.

**Формат выходных данных**

В выходной файл выведите два натуральных числа S1 и S2 – две наименьшие стоимости различных схем электроснабжения (S1 ≤ S2).

# **Текст программы на языке С#**

**Текст файла Program.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

class Program

{

static void Main()

{

StreamReader streamReader = new StreamReader("input.txt");

string[] input = streamReader.ReadLine().Split();

int n = int.Parse(input[0]);

int m = int.Parse(input[1]);

List<Edge> edges = new List<Edge>();

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

{

input = streamReader.ReadLine().Split();

int a = int.Parse(input[0]);

int b = int.Parse(input[1]);

int c = int.Parse(input[2]);

edges.Add(new Edge(a, b, c));

}

edges.Sort();

int s1 = Kruskal(n, edges);

int s2 = Kruskal(n, edges.Skip(1).ToList());

StreamWriter streamWriter = new StreamWriter("output.txt");

streamWriter.WriteLine($"{s1} {s2}");

streamWriter.Close();

}

static int Kruskal(int n, List<Edge> edges)

{

int[] parent = new int[n + 1];

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

parent[i] = i;

int cost = 0;

foreach (Edge edge in edges)

{

if (parent[edge.A] != parent[edge.B])

{

cost += edge.C;

int oldParent = parent[edge.B];

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

{

if (parent[i] == oldParent)

parent[i] = parent[edge.A];

}

}

}

return cost;

}

}

class Edge : IComparable<Edge>

{

public int A { get; }

public int B { get; }

public int C { get; }

public Edge(int a, int b, int c)

{

A = a;

B = b;

C = c;

}

public int CompareTo(Edge other)

{

return C.CompareTo(other.C);

}

}

**Текст файла input.txt:**

5 8

1 3 75

3 4 51

2 4 19

3 2 95

2 5 42

5 4 31

1 2 9

3 5 66

**Текст файла output.txt:**

110 176