Факультет компьютерных технологий

Лабораторная работа №3

«Современные платформы программирования»

Вариант №10

Выполнил: студент гр. 981063 Ефименко Павел Викторович

Проверил: Воробей В.А.

Минск 2021

Условие:

Решить задачу, в соответствии с вариантом подгруппы. В качестве среды разработки использовать IntelliJ IDEA. В разработке алгоритма использовать строго типизированный объектно-ориентированный язык программирования Java. 10. Система двусторонних дорог такова, что для любой пары городов можно указать соединяющий их путь. Найдите такой город, сумма расстояний от которого до остальных городов минимальна.

Код:

using System;

namespace Laba\_3

{

class Program

{

private const int N = 4;

static void Main(string[] args)

{

var A = Helper.FillRandomAdjacencyMatrix(N, N);

var vMin = 0;

var TS = int.MaxValue;

A.Print();

// Алгоритм Флойда-Уоршелла

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

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

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

A[i,j] = Math.Min(A[i,j], A[i,k] + A[k,j]);

// Ищем вершину с минимальной суммой

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

{

var S = 0;

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

S += A[i,j];

if (S < TS)

{

TS = S;

vMin = i;

}

}

Console.WriteLine($"вывод города с минимальным расстоянием (нумерация с 0)\n{vMin}");

}

}

}

using System;

using System.Diagnostics;

using System.Linq;

using System.Threading;

namespace Laba\_3

{

public static class Helper

{

public static char[,] ConvertToMatrix(this char[] arr, int m, int n)

{

if (arr.Length != m \* n)

{

throw new ArgumentNullException();

}

char[,] matrix = new char[m, n];

Buffer.BlockCopy(arr, 0, matrix, 0, arr.Length \* sizeof(char));

return matrix;

}

public static char[] Shuffle(this char[] arr)

{

var rnd = new Random(DateTime.Now.Millisecond + Thread.CurrentThread.ManagedThreadId);

return arr.OrderBy(item => rnd.Next()).ToArray();

}

public static char[,] FiilByMN(this char[,] m, int M, int N)

{

var matrix = new char[M, N];

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

{

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

{

matrix[i, j] = m[i, j];

}

}

return matrix;

}

public static void Print<T>(this T[,] matrix)

{

for (int i = 0; i < matrix.GetLength(0); i++)

{

for (int j = 0; j < matrix.GetLength(1); j++)

{

Console.Write($"{matrix[i,j]}\t");

}

Console.WriteLine();

}

}

public static int[,] RandomFillMatrix(int n, int m, int minValue, int maxValue)

{

var res = new int[n, m];

var rnd = new Random();

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

{

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

{

res[i, j] = rnd.Next(minValue, maxValue);

}

}

return res;

}

public static int[,] FillRandomAdjacencyMatrix(int n, int m)

{

var res = new int[n, m];

var rnd = new Random(DateTime.UtcNow.Millisecond / Process.GetCurrentProcess().Id);

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

{

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

{

res[i, j] = rnd.Next(0, 20);

if (res[i, j] == 0)

{

res[i, j] = int.MaxValue;

}

}

}

return res;

}

}

}

Результат:

|  |  |
| --- | --- |
|  |  |