# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ТАРАСА ШЕВЧЕНКА



## ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №1**

# з курсу

**«Проектування та аналіз обчислювальних алгоритмів»**

*Студента 2 курсу*

*групи ПП-21 спеціальності 122 «Комп'ютерні науки» ОП «Прикладне програмування»*

%username%

*Викладач:*

к.ф.-м.н., доц. Шолохов О.В.

## Київ – 202

**1.Назва роботи**

Пошук найбільшої та найменшої дистанції між двома точками з множини точок

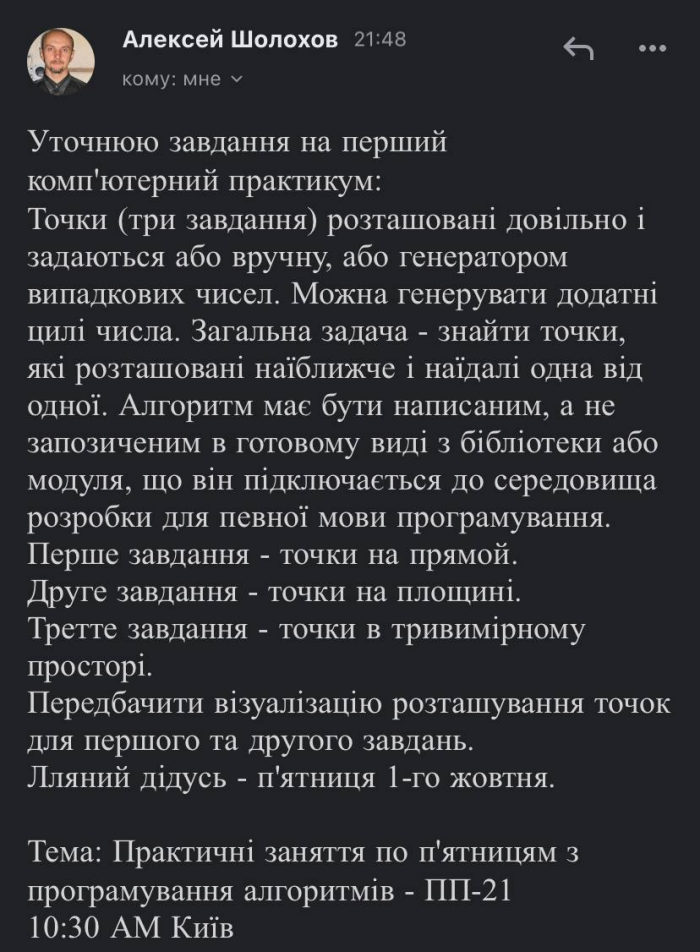
1. **Тема роботи**

Пошук найбільшої та найменшої дистанції між двома точками з множини точок

1. **Мета роботи**

Створити алгоритм для знаходження найбільшої та найменшої дистанції між парою точок у одно-, двох- та трьохвимірному просторі

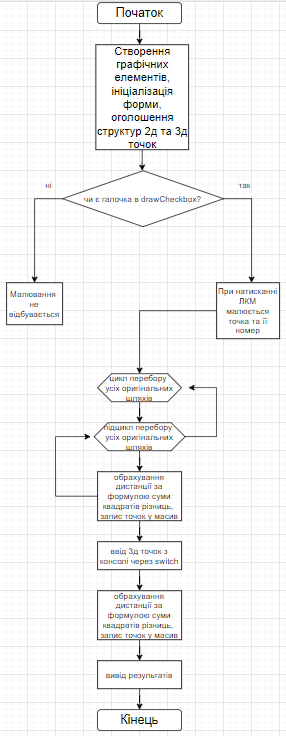
1. **Умова завдання**



1. **Рішення**

Створюємо графічні елементи, ініціалізуємо форму. Оголошуємо структури 2Д та 3Д точок, масиви вищезгаданих точок. Описуємо функцію, що відображає позицію курсору (надсилає координати курсору елементу statusLabel) та функцію, що малює на екрані точки (за умови, що чекбокс Draw має галочку при натисканні лівої кнопки миші на позиції курсору +10 пікселів малюється точка та ії номер). Після цього описуємо функції для підрахунку дистанції між точками на площині (дистанція дорівнює квадратному кореню з суми квадратів різниць відповідних кординат другої та першої точок) та пошуку усіх таких дистанцій (вищезгадана функція викликається у підциклі циклу, що перебирає всі оригінальні дистанції між точками шляхом зменшення первинного обмеження на 1, а вторинного - на q + 1), а також мінімальної та максимальної з них (шляхом сортування масиву з усіма дистанціями). Наступними описуємо функції для створення точок у трьохвимірному просторі (функція являє собою меню на основі switch, загорнутого в цикл while, що перевіряє змінну switch`а на нерівність трьом (себто, умові виходу зі switch); switch має три case, не враховуючи default: перший викликається вводом одиниці в консоль та виводить запрошення для вводу трьох кординат, кожну з яких записує в одиницю структури current3DPoint, яку в свою чергу додає до масиву 3д точок; другий case виводить на екран усі 3д точки з їх номерами, після чого рахує кожну оригінальну дистанцію між 3д точками (дистанція дорівнює квадратному кореню з суми квадратів різниць відповідних кординат другої та першої точок, а оригінальність кожної дистанції досягається шляхом зменшення первинного обмеження на 1, а вторинного - на q + 1) та виводить її на екран разом з парою відповідних точок, після чого сортує масив дистанцій та виводить максимальну і мінімальну з них.

Блок-схема:



Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace TC\_91

{

public partial class Form1 : Form

{

Graphics gr; Pen p; Point cursor;

int k = 0;

Point[] points = new Point[50];

public Form1()

{

InitializeComponent();

gr = this.CreateGraphics();

p = new Pen(Color.Black, 3);

}

struct Point3D

{

public int x;

public int y;

public int z;

}

Point3D[] point3Ds = new Point3D[50];

private void Form1\_Load(object sender, EventArgs e)

{

}

private void Form1\_MouseMove(object sender, MouseEventArgs e)

{

cursor = this.PointToClient(Cursor.Position);

cursorLocation.Text = "X: " + cursor.X + " Y: " + cursor.Y;

}

int pointCounter = 0;

public void Form1\_Click(object sender, EventArgs e)

{

if(drawCheckBox.Checked == true)

{

gr.DrawEllipse(p, cursor.X - 10, cursor.Y - 10, 1, 1);

points[k++] = new Point(cursor.X, cursor.Y);

pointsList.Items.Add("X: " + cursor.X + " Y: " + cursor.Y);

pointCounter++;

}

}

public double distanceCalculator(Point firstPoint, Point secondPoint)

{

double distance = Math.Sqrt(Math.Pow((secondPoint.X - firstPoint.X), 2) + Math.Pow((secondPoint.Y - firstPoint.Y), 2));

return distance;

}

private void submitButton\_Click(object sender, EventArgs e)

{

List<double> distances = new List<double>();

//distances.Add(12);

for (int q = 0; q < pointCounter - 1; q++)

{

for (int w = 0; w < pointCounter - q - 1; w++)

{

double currentDistance = distanceCalculator(points[q], points[w + q + 1]);

distances.Add(currentDistance);

}

}

foreach (double dist in distances)

{

Console.WriteLine(dist);

Console.ReadKey();

}

maximumTextBox.Text = Convert.ToString(distances.Max());

minimumTextBox.Text = Convert.ToString(distances.Min());

}

private void button1\_Click(object sender, EventArgs e)

{

int switchFlag = 1;

int points3DCounter = 0;

while (switchFlag != 3)

{

switch (switchFlag)

{

case 1:

{

Console.WriteLine("Please enter X, Y and Z of the first 3D point:");

Point3D current3DPoint = new Point3D();

Console.WriteLine("Enter X:");

current3DPoint.x = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Now enter Y:");

current3DPoint.y = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Now enter Z:");

current3DPoint.z = Convert.ToInt32(Console.ReadLine());

point3Ds[points3DCounter] = current3DPoint;

points3DCounter++;

Console.WriteLine("Enter 1 to continue entering 3D points, 2 to submit, or 3 to exit:");

switchFlag = Convert.ToInt32(Console.ReadLine());

}

break;

case 2:

Console.WriteLine("3D points:");

for (int q = 0; q < points3DCounter; q++)

{

Console.Write("Point " + q + " X: " + point3Ds[q].x + ", Y: " + point3Ds[q].y + ", Z: " + point3Ds[q].z);

Console.WriteLine("");

}

Console.WriteLine("Points submitted, enter 3 to exit:");

switchFlag = Convert.ToInt32(Console.ReadLine());

double distance3DCalculator(Point3D firstPoint, Point3D secondPoint)

{

double distance = Math.Sqrt(Math.Pow((secondPoint.x - firstPoint.x), 2) + Math.Pow((secondPoint.y - firstPoint.y), 2) + Math.Pow((secondPoint.z - firstPoint.z), 2));

return distance;

}

List<double> distances3d = new List<double>();

for (int q = 0; q < points3DCounter - 1; q++)

{

for (int w = 0; w < points3DCounter - q - 1; w++)

{

double currentDistance = distance3DCalculator(point3Ds[q], point3Ds[w + q + 1]);

distances3d.Add(currentDistance);

}

}

maximum3D.Text = Convert.ToString(distances3d.Max());

minimum3D.Text = Convert.ToString(distances3d.Min());

break;

case 3:

break;

default:

break;

}

}

}

}

}

Результат роботи програми:

[screenshot]

1. **Висновки**

В результаті виконання даної лабораторної роботи я оволодів практичними навичками створення алгоритмів для виконання типових завдань на пошук максимальної та мінімальної дистанції між двома точками серед множини точок у одно-, двох- та трьохвимірному просторах. Вважаю дану лабораторну роботу виконаною в повному обсязі.