Lucrare de laborator

-Parcurgerea in latime-

Descriere program

Aplicatia de fata, constituie o reprezentare grafica a parcurgerii in latime a unui graf neorientat .

Pentru realizarea acestei aplicatii, am generat random o matrice de adiacenta, precum si coordonatele nodurilor existente, acestea din urma(coordonatele) fiind inregistrate intr-o lista. Pentru parcurgerea grafului, am folosit algoritmul classic de parcurgere in latime, pe care l-am transpus intr-un timer. La generarea fiecarei matrici, se poate selecta , prin intermediul unui ComboBox, nodul de unde sa se inceapa parcurgerea, iar nodurile rezultate in urma parcurgerii sunt afisate intr-un TextBox

Codul sursa

```
public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        int n;
        int[,] a;
        int min = 5;
        int max = 15;
        int x, i, j;

        public bool verif( List<Point> l, Point p)
        {

            foreach(Point elem in l)
            {
                if (Math.Sqrt(Math.Pow((p.X - elem.X), 2) + Math.Pow((p.Y - elem.Y), 2)) < 50)
                    return false;
            }
            foreach(Point p1 in l)
            {
                foreach(Point p2 in l)
                {
                    if (p1 != p2)
                    {
                        if (p.X * p1.Y + p1.X * p2.Y + p.Y * p2.X - p.X * p2.Y - p2.X * p1.Y - p.Y *
p1.X == 0)
                            return false;
                    }
                }
            }
            return true;
        }
        bool gen = false;
        private void Form1_Load(object sender, EventArgs e) { }
        private void comboBox1_SelectedIndexChanged(object sender, EventArgs e) { }
        private void button1_Click(object sender, EventArgs e)
        {
            Point[] crd = new Point[max];
            gen_Mat_Adiacenta();
```

```
            Random coords = new Random();
            List<Point> result = new List<Point>();//generez punctele(coordonatele) random, astefel
incat sa fie diferite intre ele-lista de puncte

            for (i = 0; i < n; i++)
            {
                Point pct_curent = new Point();
                do
                {
                  pct_curent = new Point(coords.Next(5, this.panel1.Width - 25), coords.Next(5,
this.panel1.Height-25));
                } while (verif( result,pct_curent)==false);

                result.Add(pct_curent);
            }

            i = 0;
            foreach(Point pct in result)
            {
                crd[i] = pct;
                i++;
            }

            //desenez nodurile grafului, in centrul carora este scris numarul nodului.
            Graphics g = panel1.CreateGraphics();
            g.Clear(this.BackColor);

            for (i = 0; i < n; i++)
            {
                string text = Convert.ToString(i);
                RectangleF rectF1 = new RectangleF(crd[i], new Size(20, 20));
                g.DrawRectangle(new Pen(Color.Black), new Rectangle(crd[i], new Size(20, 20)));
                g.DrawString(text, new Font("Arial", 9, FontStyle.Bold, GraphicsUnit.Point),
Brushes.Black, rectF1);
                g.DrawRectangle(Pens.Black, Rectangle.Round(rectF1));
                    for (j = i + 1; j < n; j++)
                    {
                        if (a[i, j] == 1)
                        {
                            g.DrawLine(new Pen(Color.Black), crd[i], crd[j]);
                        }
                    }
            }
            //generez numerele(capetele de noduri) de unde voi incepe parcurgerea.

            if (gen == true) comboBox1.Items.Clear();//daca exista alte puncte generate anterior , le
sterg
                for (i = 0; i < n; i++)
                {
                    comboBox1.Items.Add(i);
                    gen = true;
                }
        }

        public void gen_Mat_Adiacenta()//generez random matricea de adiacenta
        {
            Random randNum = new Random();
            n = randNum.Next(min, max);
            a = new int[n, n];

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

```
            int procent = randNum.Next(0, 101);
            if (procent % 5 == 0)
                a[j, i] = a[i, j] = 1;
        }
    }
    afis_matrice();
}
public void afis_matrice()//afisez in consola matricea de adiacenta
    //asta e doar de verificare ca sa stiu daca am facut corect
{
    int z, q;
    for (z = 0; z < n; z++)
    {
        for (q = 0; q < n; q++)
        {
            Console.Write(a[z, q] + " ");
        }
        Console.WriteLine();
    }
}
int[] viz = new int[30];
int[] coada = new int[30];
int p = 0, u = 0, el;

private void panel1_Paint(object sender, PaintEventArgs e)
{

}

public void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    for (i = 0; i < n; i++) viz[i] = 0;
    string s = this.comboBox1.GetItemText(this.comboBox1.SelectedItem);
    label1.Text = s;
    x = Convert.ToInt32(s);
    j = 0;
    viz[x] = 1;
    Coada.Add(x);
    timer1.Enabled = true;
    coada[0] = x;
}

List<int> Coada = new List<int>();
private void timer1_Tick(object sender, EventArgs e)
{

    if (Coada.Count != 0)
    {
        el = Coada.First();
        bool ok = false;
          for (j=0;j<n;j++)

        {
            if ((a[el, j] == 1) && (viz[j] == 0))
            {
                Coada.Add(j);
                this.label1.Text += Convert.ToString(j);
                viz[j] = 1;
                ok = true;
            }
            if (ok == true)
            {
                break;
```

```
                    }

                }

                if (j == n)
                    Coada.Remove(el);

            }
            else
                timer1.Enabled = false;
            checkBox1.Checked = false;

        }
    }
}
```

La rularea aplicatie , va fi afisat urmatorul ecran: