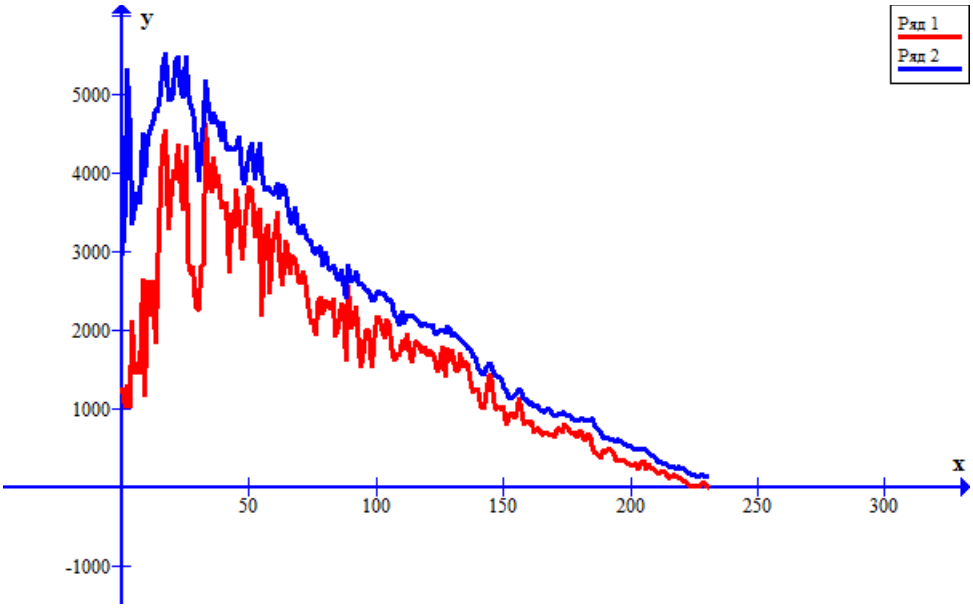


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;

namespace Siit_3
{
   class generation
   {
```

```
static public int numChromo = 20;
List<int[]> gens;
List<int> fitness { get; }
List<double> sharedFitness;
List<float> probability { get; }
List<int> chromSelect;
List<int[]> location;
List<int> arround;
public int sousage_x1 = 800, sousage_x2 = 800, sousage_x3 = 200, sousage_x4 = 200;
public int sousage_y1 = 800, sousage_y2 = 200, sousage_y3 = 800, sousage_y4 = 200;
public double averagefitness = 0f;

Random mutat = new Random();
int rando = 0;

public generation()
{
   gens = new List<int[]>();
   fitness = new List<int>();
   probability = new List<float>();
   chromSelect = new List<int>();
   location = new List<int[]>();
   sharedFitness = new List<double>();
   arround = new List<int>();

   for (int j = 0; j < numChromo; j++)
   {
      int[] gen = new int[1000];
      location.Add(new int[] { 500,500});
      gens.Add(gen);
      fitness.Add(0);
      arround.Add(0);
      sharedFitness.Add(0);
      probability.Add(0f);
      chromSelect.Add(0);
   }
}

public generation(List<int[]> new_gens)
{
   gens = new List<int[]>();
   fitness = new List<int>();
   probability = new List<float>();
   chromSelect = new List<int>();
   location = new List<int[]>();
   sharedFitness = new List<double>();
   arround = new List<int>();
   gens = new_gens;
   for (int j = 0; j < numChromo; j++)
   {
      location.Add(new int[] { 500, 500 });
      fitness.Add(0);
```

```
            sharedFitness.Add(0);
            arround.Add(0);
            probability.Add(0f);
            chromSelect.Add(0);
        }
    }

    public void randomize()
    {
        Random rand = new Random();
        for (int i = 0; i < numChromo; i++)
        {
            for (int j = 0; j < 1000; j++)
            {

                gens[i][j] = rand.Next() % 4 + 1;
            }
        }
    }
    public void setFitness()
    {
        for (int i = 0; i < numChromo; i++)
        {
            for (int j = 0; j < 1000; j++)              //1 - up || y++
            {                                       //2 - right || x++
                if (gens[i][j] == 1) location[i][1]++;          //3 - down || y--
                else if (gens[i][j] == 2) location[i][0]++;  //4 - left || x--
                else if (gens[i][j] == 3) location[i][1]--;
                else location[i][0]--;
            }
            List<int> fit = new List<int>();
            fit.Add( Math.Abs(sousage_x1 - location[i][0]) + Math.Abs(sousage_y1 - location[i][1]));
            fit.Add(Math.Abs(sousage_x2 - location[i][0]) + Math.Abs(sousage_y2 - location[i][1]));
            fit.Add(Math.Abs(sousage_x3 - location[i][0]) + Math.Abs(sousage_y3 - location[i][1]));
            fit.Add(Math.Abs(sousage_x4 - location[i][0]) + Math.Abs(sousage_y4 - location[i][1]));
            fitness[i] = fit.Min();
        }
        for(int i = 0; i < numChromo; i++)
        {
            double sum = 0;
            for(int j =0; j < numChromo; j++)
            {
                int tmp = Math.Abs(location[i][0] - location[j][0]) + Math.Abs(location[i][1] -
location[j][1]);
                if (tmp < 50)
                {
                    arround[i]++;
                    sum += 1 - (double)tmp / 50;
                }
            }
            sharedFitness[i] = (fitness[i])*sum;
        }
```

```
}

public void setProbability()
{
   double mass = 0;
   for (int i = 0; i < numChromo; i++)
   {
      mass += sharedFitness[i];
   }
   averagefitness = mass / numChromo;
   for (int i = 0; i < numChromo; i++)
   {
      probability[i] = (float)fitness[i] / (float)mass;
   }
}
public int[] newChild()
{

   Random rand = new Random(DateTime.Now.TimeOfDay.Milliseconds + rando);
   rando++;
   if (rando == 10000000) rando = 0;
   int rand_num = rand.Next(numChromo/2);
   float sum = 0f;
   int[] chrom_1 = new int[1000], chrom_2 = new int[1000];

   //for (int i = 0; i < 100; i++)
   //{
   //    sum += probability[i] * 1000000000;
   //    if (rand_num <= sum)
   //    {
   //       chromSelect[i]++;
   //       chrom_1 = gens[i];
   //       break;
   //    }

   //}
   chrom_1 = gens[rand_num];                    // for truncate
   sum = 0f;
   rand_num = rand.Next(numChromo/2);
   //for (int i = 0; i < 100; i++)
   //{
   //    sum += probability[i] * 1000000000;
   //    if (rand_num <= sum)
   //    {
   //       chromSelect[i]++;
   //       chrom_2 = gens[i];
   //       break;
   //    }
   //}
   chrom_2 = gens[rand_num];                     // for truncate
```

```
    int[] new_chrom = new int[1000];

    //uniform crossover
    for (int i = 0; i < 1000; i++)
    {
        if (rand.Next() % 2 == 1) new_chrom[i] = chrom_1[i];
        else new_chrom[i] = chrom_2[i];
    }

    //one point crossover
    //int point = rand.Next() % 1000;
    //for (int i = 0; i < 1000; i++)
    //{
    //    if (i < point) new_chrom[i] = chrom_1[i];
    //    else new_chrom[i] = chrom_2[i];
    //}
    Mutation(new_chrom);
    return new_chrom;
}
public double bestFitness()
{
    return sharedFitness.Min();
}

public void Sort()
{
    for (int i = 0; i < numChromo - 1; i++)
    {
        bool swapped = false;
        for (int j = 0; j < numChromo - i - 1; j++)
        {
            if (sharedFitness[j] > sharedFitness[j + 1])
            {
                int[] tmp_gen = gens[j];
                gens[j] = gens[j + 1];
                gens[j + 1] = tmp_gen;

                int tmp_fit = fitness[j];
                fitness[j] = fitness[j + 1];
                fitness[j + 1] = tmp_fit;

                double tmp_shr_fit = sharedFitness[j];
                sharedFitness[j] = sharedFitness[j + 1];
                sharedFitness[j + 1] = tmp_fit;
                swapped = true;
            }

        }
        if (!swapped) break;
    }
}
```

```
        public double getAverageFit()
        {
            return averagefitness;
        }
        public void WriteTable(StreamWriter file1, StreamWriter file2)
        {
            for (int i = 0; i < numChromo; i++)
            {
                file1.WriteLine(chromSelect[i].ToString());
                file2.WriteLine(i.ToString());
            }
            file1.WriteLine();
            file1.WriteLine();
        }
        public int[] GetMaxChromo()
        {
            return gens[0];
        }
        public int[] GetChromo(int index)
        {
            return gens[index];
        }

        private void Mutation(int[] chromo)
        {
            for(int i = 0;i<1000;i++)
            {
                if(mutat.Next()%100 == 1)
                {
                    int tmp = mutat.Next() % 4 + 1;
                    if (tmp == chromo[i]) chromo[i] = (tmp + 1) % 4 +1;
                }
            }
        }
        public int getBestArround()
        {
            return arround.Min();
        }

    }
}
```

```
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;
using System.IO;


namespace Siit_3
{
   public partial class Form1 : Form
   {
      Bitmap bit = new Bitmap(1000, 1000);
      public Form1()
      {
         InitializeComponent();

         pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;


      }

      private void button1_Click(object sender, EventArgs e)
      {
         StreamWriter avgFitFile = new StreamWriter("averageFit.txt");
         StreamWriter maxFitFile = new StreamWriter("maxFit.txt");
         StreamWriter numGenFile = new StreamWriter("numGen.txt");
         StreamWriter tableFile = new StreamWriter("Table.txt");
         StreamWriter tablenum = new StreamWriter("Num.txt");
         generation old_gens = new generation();
         old_gens.randomize();
         old_gens.setFitness();
         old_gens.setProbability();
         double maxFit = 0;
         int numGeneration = 0;
         for (int j = 0; (j < 1000) && (numGeneration<250); numGeneration++)
         {
            bit = new Bitmap(1000, 1000);
            //for (int i = 0; i < 100; i++)
            //{
            //    PaintWay(old_gens.GetChromo(i));
            //}
            //MessageBox.Show("gav gav!");
            numGenFile.WriteLine(numGeneration.ToString());
            Console.WriteLine(old_gens.bestFitness() + " " + old_gens.getAverageFit() + " " +
old_gens.getBestArround());
            if (old_gens.bestFitness() == 0) break;
            List<int[]> new_tmp = new List<int[]>();
            //old_gens.Sort();                                    //for truncate
            for (int i = 0; i < generation.numChromo; i++)
            {
               new_tmp.Add(old_gens.newChild());
            }
            old_gens.WriteTable(tableFile, tablenum);
            generation new_gens = new generation(new_tmp);
```

```
        old_gens = new_gens;
        old_gens.setFitness();
        old_gens.setProbability();
        avgFitFile.WriteLine(old_gens.getAverageFit().ToString());
        maxFitFile.WriteLine(old_gens.bestFitness().ToString());
        if (old_gens.bestFitness() > maxFit)
        {
           maxFit = old_gens.bestFitness();
           j = 0;
        }
        else j++;

     }
     for (int i = 0; i < generation.numChromo; i++)
     {
        PaintWay(old_gens.GetChromo(i));
     }
     tablenum.Close();
     tableFile.Close();
     numGenFile.Close();
     avgFitFile.Close();
     maxFitFile.Close();
  }
  private void PaintWay(int[] chromo)
  {
     double location_x = 500, location_y = 500;

    bit.SetPixel(200, 200, Color.Red);
     bit.SetPixel(200, 800, Color.Red);
     bit.SetPixel(800, 200, Color.Red);
     bit.SetPixel(800, 800, Color.Red);
     Graphics graph = Graphics.FromImage(bit);
     Pen my_pen = new Pen(Color.Green);
     my_pen.Width = 10;
     graph.DrawEllipse(my_pen, 800 - 2, 800 - 2, 4, 4);
     graph.DrawEllipse(my_pen, 800 - 2, 200 - 2, 4, 4);
     graph.DrawEllipse(my_pen, 200 - 2, 800 - 2, 4, 4);
     graph.DrawEllipse(my_pen, 200 - 2, 200 - 2, 4, 4);
     foreach (int i in chromo)
     {
        if (i == 1)
        {
           location_y += 1;
           if (location_y == 1000) location_y = 1;
           bit.SetPixel((int)location_x,(int) location_y, Color.Red);
        }
        else if (i == 2)
        {
           location_x += 1;
           if (location_x == 1000) location_x = 1;
           bit.SetPixel((int)location_x, (int)location_y, Color.Red);
        }
```

```
            else if (i == 3)
            {
                location_y -= 1;
                if (location_y == 0) location_y = 999;
                bit.SetPixel((int)location_x, (int)location_y, Color.Red);
            }

            else
            {
                location_x -= 1;
                if (location_x == 0) location_x = 999;
                bit.SetPixel((int)location_x, (int)location_y, Color.Red);
            }

        }
        pictureBox1.Image = bit;
    }
  }
}
```