****

**รายงาน**

**การค้นหาเส้นทางโดยวิธี A\* Search**

จัดทำโดย

นายบัณฑิต อ่อนมา 541733023005-4

นายจตุรงค์ คำทอง 541733023012-0

นายธนาธิป โพธิ์ธรรม 541733023027-8

นางสาวราตรี ฉาสันทียะ 541733023032-8

**เสนอ**

**อาจารย์ ธรรมกร ครองไตรภพ**

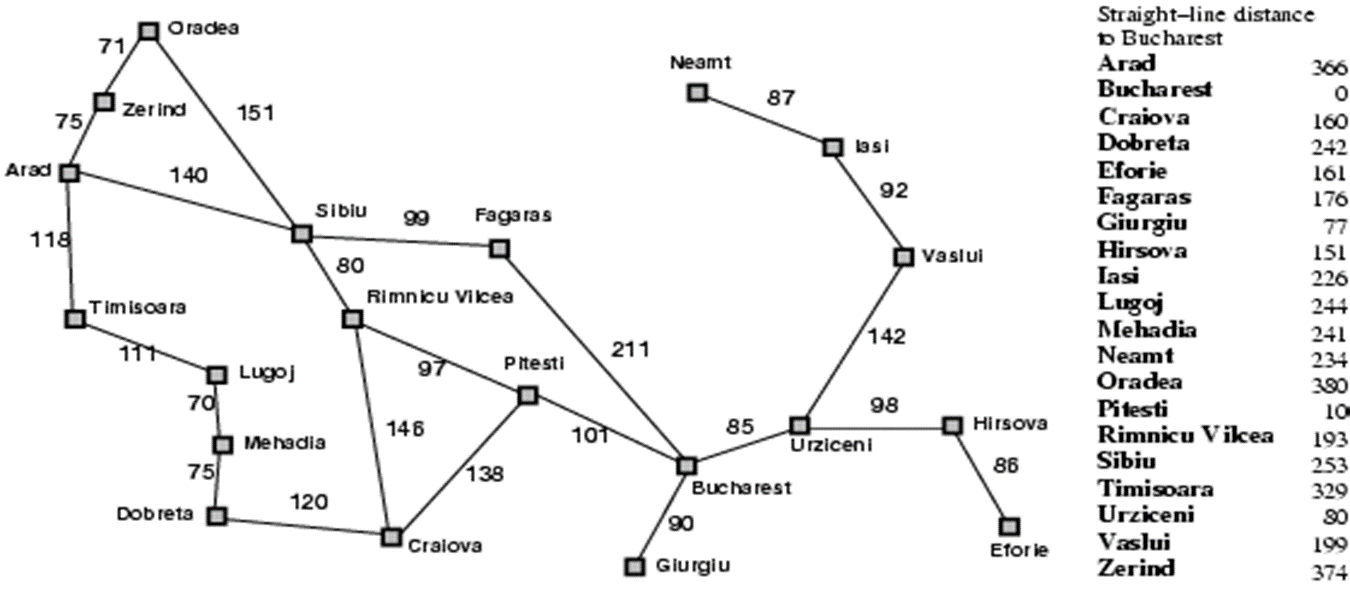
รายงานนี้เป็นส่วนหนึ่งของวิชา  ปัญญาประดิษฐ์ (Artificial Intelligence)

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์และสถาปัตยกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน ปีการศึกษา  2556

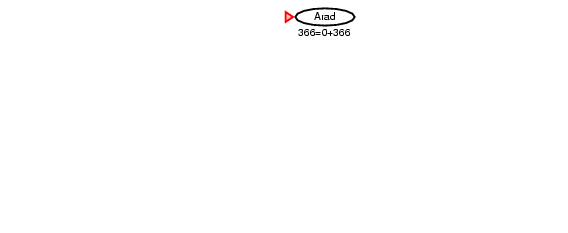
เขียนโปรแกรมแสดงขั้นตอนการค้นหาเส้นทางจากเมืองใดๆไปยังเมือง Bucharest

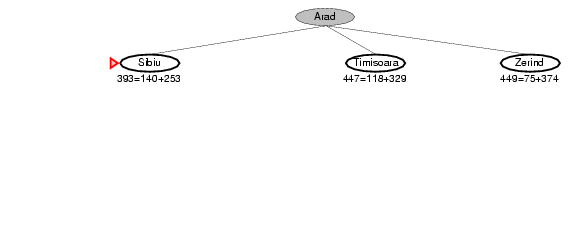
โดยใช้วิธีการค้นหาแบบ A Star Search ?

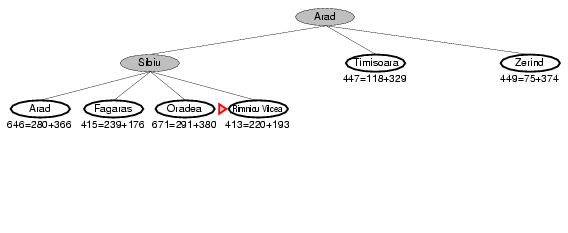


**ตัวอย่างการค้นหาแบบ A\* Search**

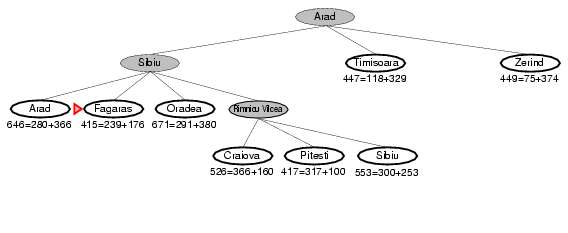
ตัวอย่างการค้นหาจากเมือง Arad ไปยังเมือง Bucharest

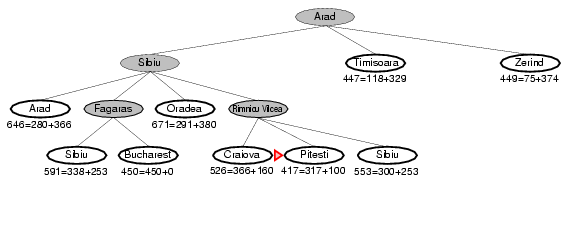
**1.**

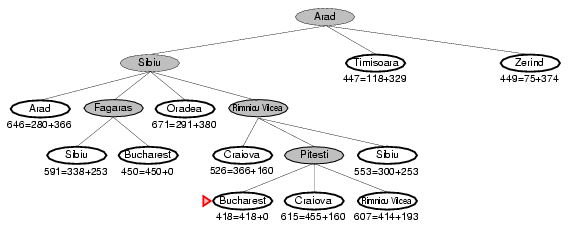
**2.**

****

**3.**

**4.**

**5.**

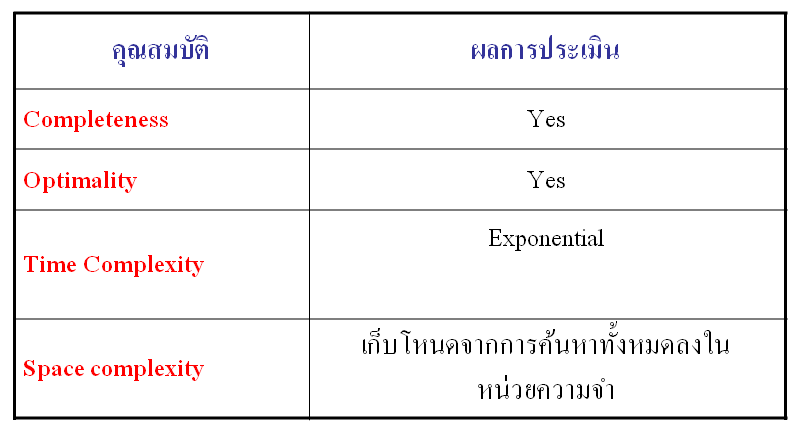
****

**6.**

**ทฤษฏี**

* เป็นวิธีการค้นหาที่ถูกพัฒนาขึ้นเพื่อแก้ปัญหาที่เกิดขึ้นใน Greedy Best First Search
* โดยนำข้อมูลเข้ามาพิจารณาเพิ่มเติม คือ ข้อมูลทรัพยากรที่ใช้ตั้งแต่ตำแหน่งเริ่มต้นจนถึงตำแหน่งที่กำลังพิจารณาขณะนั้น หากนำมาเขียนเป็นสัญลักษณ์เป็น *g*(n) จะหมายถึง “ข้อมูลทรัพยากรที่ใช้ตั้งแต่เริ่มต้นจนถึงโหนด n”
* ดังนั้น ฟังก์ชัน Evaluation สามารถเขียนอยู่ในรูปแบบสมการดังนี้
  + *f*(n) = *g*(n) + *h*(n)
* การค้นหาจะพิจารณาโหนดที่มีค่า Heuristic น้อย จนกว่าจะพบเป้าหมายที่สั้นที่สุด

**คุณสมบัติของการค้นหาแบบ A\* Search**

****

**Flow Chart**

ไม่ใช่

ใช่

เริ่ม

เลือกเมืองต้นทาง

ค้นหาเมืองที่ติดกัน

คำนวณระยะทางติดกันแต่ละเมือง

เลือกเมืองที่มีระยะทางน้อยที่สุดของเมืองที่ติดกัน

เมืองที่เลือกมีระยะทางน้อยกว่าเมืองที่ไม่ถูกเลือก

ค้นหาเมืองที่ยังไม่ถูกเลือกและมีค่าน้อยที่สุดและน้อยกว่าเมืองที่เลือกในปัจจุบัน

ค้นหาเมืองเป้าหมายและมีระยะทางน้อยที่สุด

ใช่

ตรวจสอบระดับ

ตรวจสอบจำนวนครั้งในการวาด

วาดเมืองต้นทาง

ตรวจสอบเมืองแต่ละระดับ ว่ามีการวาดกี่ครั้ง

แบ่งพื้นที่แต่ละระดับ

เลือกตำแหน่งการวาดในระดับที่วาด

วาดเมืองและเส้นเชื่อมต่อเมือง

ลากเส้นที่ทำการเลือกในแต่ละครั้ง

ตรวจสอบการวาดว่าครบหรือไม่

จบการทำงาน

ไม่ใช่

**โค้ดโปรแกรมการค้นหาแบบ A\* Search**

**Form1**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace A\_Star\_Search

{

public partial class Form1 : Form

{

private Dictionary<string, string> arr\_short\_name = new Dictionary<string, string>()

{

{"Arad","a"}, {"Bucharest","b"}, {"Craiova","c"}, {"Dobreta","d"},

{"Eforie","e"}, {"Fagaras","f"}, {"Giurgiu","g"}, {"Hirsova","h"},

{"Iasi","i"}, {"Lugoj","l"}, {"Mehadia","m"}, {"Neamt","n"},

{"Oradea","o"}, {"Pitesti","p"}, {"Rimnicu","r"}, {"Sibiu","s"},

{"Timisoara","t"}, {"Urziceni","u"}, {"Vaslui","v"}, {"Zerind","z"},

};

string[] city = new string[] { "Arad", "Bucharest", "Craiova", "Dobreta", "Eforie", "Fagaras", "Giurgiu", "Hirsova", "Iasi", "Lugoj",

"Mehadia", "Neamt", "Oradea", "Pitesti", "Rimnicu Vilcea", "Sibiu", "Timisoara", "Urziceni", "Vaslui", "Zerind"};

int[,] travel = new int[31, 17];//(เมือง,ระยะทางที่ห่างจากเมือง Bucharest,ระยะทางสะสม,ระยะทางรวม)X4 ,ระดับ

int[,] select\_city = new int[31, 4];//หมายเลขเมือง,ตำแหน่ง row ในอาร์เรย์ travel,ตำแหน่ง column ในอาร์เรย์ travel,ระยะทาง

int[,] dis\_city = new int[20, 10]{

{0,366,15,140,16,118,19,75,0,0},

{1,0,5,211,6,90,13,101,17,85},

{2,160,3,120,13,138,14,146,0,0},

{3,242,2,120,10,75,0,0,0,0},

{4,161,7,86,0,0,0,0,0,0},

{5,176,1,211,15,99,0,0,0,0},

{6,77,1,90,0,0,0,0,0,0},

{7,151,4,86,17,98,0,0,0,0},

{8,226,11,87,18,92,0,0,0,0},

{9,244,10,70,16,111,0,0,0,0},

{10,241,3,75,9,70,0,0,0,0},

{11,234,8,87,0,0,0,0,0,0},

{12,380,15,151,19,71,0,0,0,0},

{13,100,1,101,2,138,14,97,0,0},

{14,193,2,146,13,97,15,80,0,0},

{15,253,0,140,5,99,12,151,14,80},

{16,329,0,118,9,111,0,0,0,0},

{17,80,1,85,7,98,18,142,0,0},

{18,199,8,92,17,142,0,0,0,0},

{19,374,0,75,12,71,0,0,0,0}

};//หมายเลขเมือง,ระยะทางที่ห่างจากเมือง Bucharest,หมายเลขเมืองที่อยู่ติดกัน <= 4,ระยะทางระหว่างเมือง <= 4

int start, speed;

Form2 Answer\_form;

public Form1()

{

InitializeComponent();

}

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

{

//กำหนดค่า Default ที่เมือง Arad

cbbStart.SelectedIndex = 0;

((PictureBox)Floor.Panel1.Controls["town\_a"]).BorderStyle = BorderStyle.FixedSingle;

Floor.Panel1.Controls["town\_a"].Visible = true;

Floor.Panel1.Controls["town\_a"].BackgroundImage = global::A\_Star\_Search.Properties.Resources.\_true;

}

private void cbbStart\_SelectedIndexChanged(object sender, EventArgs e)

{

for (int i = 0; i < Floor.Panel1.Controls.Count; i++)

{

if (Floor.Panel1.Controls[i].Name.Contains("town\_"))

{

Floor.Panel1.Controls[i].Visible = true;

Floor.Panel1.Controls[i].BackgroundImage = null;

((PictureBox)Floor.Panel1.Controls[i]).BorderStyle = BorderStyle.FixedSingle;

}

}

//กำหนดให้ ตำแหน่ง Bucharest เป็นรูปบ้าน

((PictureBox)Floor.Panel1.Controls["town\_b"]).BorderStyle = BorderStyle.FixedSingle;

Floor.Panel1.Controls["town\_b"].Visible = true;

Floor.Panel1.Controls["town\_b"].BackgroundImage = global::A\_Star\_Search.Properties.Resources.home;

//if (cbbStart.SelectedIndex == 0)

//{

// return;

//}

// กำหนดให้เมื่อเลือกเมือง จะนำชื่อที่ได้ไปแปลงเป็นอักษรย่อ จาก Dictionary ที่กำหนด

// และกำหนดให้มีรูปเครื่องหมายถูกที่ตำแหน่งเมืองนั้นๆ

string town = arr\_short\_name[cbbStart.SelectedItem.ToString()];

Floor.Panel1.Controls["town\_" + town].Visible = true;

Floor.Panel1.Controls["town\_" + town].BackgroundImage = global::A\_Star\_Search.Properties.Resources.\_true;

//btnSearch.Focus();

}

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

{

if (cbbStart.SelectedIndex < 1)

{

start = cbbStart.SelectedIndex;

}

else

{

// +1 เนื่องจาก ตำแหน่งที่ 1 เป็นเมือง Bucharest ที่ไม่มีอยู่ใน List รายการ

start = cbbStart.SelectedIndex + 1;

}

a\_search(start, 1);

speed = (10 - Convert.ToInt32(numericUpDown1.Value.ToString())) \* 150;

Answer\_form = new Form2(travel, select\_city, speed);

Answer\_form.ShowDialog();

}

private void a\_search(int start\_, int destination\_)//ค้นหาแบบ A\_star

{

//////////////////////Clear ค่าใน Array /////////////////////////////////

Array.Clear(select\_city, 0, select\_city.Length);

Array.Clear(travel, 0, travel.Length);

//////////////////////////ประกาศตัวแปร กำหนดค่าในตัวแปร/////////////////////////////////////////////

int[] step\_traval = new int[17];

int[] choose\_now, choose\_new = new int[4];

int back = 0;

select\_city[0, 0] = start\_;//เมืองที่เลือก

select\_city[0, 1] = 0;//ตำแหน่ง row ในอาร์เรย์ travel

select\_city[0, 2] = 0;//ตำแหน่ง column ในอาร์เรย์ travel

select\_city[0, 3] = dis\_city[start\_, 1] + 0;//ระยะทางรวม

travel[0, 0] = start\_;//เมืองที่เลือก

travel[0, 1] = dis\_city[start\_, 1];//ระยะทางที่ห่างจากเมือง Bucharest

travel[0, 2] = 0; //ระยะทางสะสม

travel[0, 3] = dis\_city[start\_, 1] + 0;//ระยะทางรวม

//////////////////////////////วนรอบทำการค้นหา/////////////////////////////////////////

for (int i = 1; i < 31; i++)

{

//////////////////////////////ค้นหาเมืองที่จะไปได้///////////////////////////////////

step\_traval = find\_city(select\_city[i - 1, 0], i - 1 - back);//หาเมืองที่จะไป,ระยะทางจาก Bucharest,ระยะทางสะสม,ระยะทางรวม

step\_traval[16] = chk\_lv(i - back);//ใส่ค่าระดับ

for (int j = 0; j < 17; j++)

{

travel[i, j] = step\_traval[j];

}

choose\_now = choose\_city(step\_traval, i);//หมายเลขเมือง,ตำแหน่ง row ในอาร์เรย์ travel,ตำแหน่ง column ในอาร์เรย์ travel,ระยะทาง

choose\_new = sch\_dis\_city\_less(choose\_now[3], i);

//////////////////////เปรียบเทียบหาค่าที่น้อยสุดเพื่อนำไปค้นหาเมืองที่จะไปได้/////////////////////

if (choose\_now[3] > choose\_new[3] && choose\_new[3] != 0)//ถ้ามีเมืองที่มีระยะทางสะสมน้อยกว่าเมืองที่เลือกในปัจจุบัน

{

select\_city[i, 0] = choose\_new[0];//เมืองที่เลือก

select\_city[i, 1] = choose\_new[1];//ตำแหน่ง row ในอาร์เรย์ travel

select\_city[i, 2] = choose\_new[2];//ตำแหน่ง column ในอาร์เรย์ travel

select\_city[i, 3] = choose\_new[3];//ระยะทางรวม

back = i - choose\_new[1];

}

else

{

select\_city[i, 0] = choose\_now[0];//เมืองที่เลือก

select\_city[i, 1] = choose\_now[1];//ตำแหน่ง row ในอาร์เรย์ travel

select\_city[i, 2] = choose\_now[2];//ตำแหน่ง column ในอาร์เรย์ travel

select\_city[i, 3] = choose\_now[3];//ระยะทางรวม

back = 0;

}

//select\_city[i, 1] = step\_traval[16];

//////////////////////////เงื่อนไขในการหยุดค้นหา//////////////////////////////

if (select\_city[i, 0] == destination\_ && choose\_new[3] == 0)

{

i = 32;

for (int k = 0; k < 17; k++)

{

select\_city[k, 1] = travel[k, 16];//เปลี่ยนจากค่าตำแหน่ง row ในอาร์เรย์ travel เป็นระดับ

}

}

}

}

private int[] find\_city(int source\_city, int step\_for\_sch)//หาเมืองที่จะไป,ระยะทางจาก Bucharest,ระยะทางสะสม,ระยะทางรวม

{

int[] step\_traval\_ = new int[17];

int count = 0;

int index\_of\_source\_city = find\_index\_no\_city(source\_city, step\_for\_sch); //ลำดับของเมืองกำหนด ที่อยู่ในอาร์เรย์ travel ในชั้นที่ส่งมา

for (int i = 2; i < 10; i += 2) //หาเมืองที่จะไปได้

{

if (dis\_city[source\_city, i + 1] != 0)//ถ้าระยะทางระหว่างเมืองมากกว่าในตำแหน่งที่หาอยู่ไม่เป็น 0

{

step\_traval\_[count] = dis\_city[source\_city, i];//หมายเลขเมือง

count++;

step\_traval\_[count] = dis\_city[step\_traval\_[count - 1], 1];//ระยะทางที่ห่างจากเมือง Bucharest ของเมืองที่สามารถไปได้

count++;

step\_traval\_[count] = dis\_city[source\_city, i + 1] + travel[step\_for\_sch, index\_of\_source\_city + 2];//ระยะทางสะสม

count++;

step\_traval\_[count] = dis\_city[step\_traval\_[count - 3], 1] + step\_traval\_[count - 1];//ระยะทางรวม

count++;

}

}

return step\_traval\_;

}

private int[] choose\_city(int[] step\_travel\_, int step\_for\_sch)//หาเมืองที่เลือก, ตำแหน่งในอาร์เรย์ travel, ระยะทาง

{

int[] choose = new int[4];//หมายเลขเมือง,ตำแหน่ง row ในอาร์เรย์ travel,ตำแหน่ง column ในอาร์เรย์ travel,ระยะทาง

int less = 5000;

for (int i = 3; i < 16; i += 4) //หาเมืองที่มีระยะทางรวมน้อยที่สุด

{

if (step\_travel\_[i] != 0)

{

if (step\_travel\_[i] < less)

{

less = step\_travel\_[i];

choose[0] = step\_travel\_[i - 3];

choose[1] = step\_for\_sch;

choose[2] = (i - 3) / 4;

choose[3] = less;

}

}

}

return choose;

}

private int find\_index\_no\_city(int soucer\_city, int step\_for\_sch)//หาลำดับของเมืองกำหนด ที่อยู่ในอาร์เรย์ travel ในชั้นที่ส่งมา

{

int index\_ = 0;

for (int i = 0; i < 16; i += 4)

{

if (soucer\_city == travel[step\_for\_sch, i] && travel[step\_for\_sch, i + 3] != 0)

{

index\_ = i;

}

}

return index\_;

}

private int[] sch\_dis\_city\_less(int choose\_distance, int sum\_choose)//หาเมืองที่มีระยะทางรวมน้อยกว่าค่าที่ส่งมา และยังไม่เคยถูกเลือกมาก่อน

{

int[] choose = new int[4];//เมือง,ตำแหน่งในอาร์เรย์ travel,ระยะทาง

int less = 5000;

int chk;

for (int i = 1; i < sum\_choose; i++)//หาเมืองที่มีระยะทางรวมน้อยที่สุด

{

for (int j = 3; j < 16; j += 4)//หาเมืองที่มีระยะทางรวมน้อยที่สุด

{

chk = 0;

if (travel[i, j] != 0)//ถ้าระยะทางรวมน้อยที่สุดไม่เป็น 0

{

if (travel[i, j] < less && travel[i, j] < choose\_distance)//ถ้าระยะทางรวมน้อยกว่าค่า less และน้อยกว่าค่า choose\_distance

{

for (int k = 1; k <= sum\_choose; k++)

{

if (travel[i, j] == select\_city[k, 3])//ถ้าระยะทางรวมไม่ตรงกับเมืองที่เคยเลือก

{

chk = 1;

}

}

if (chk == 0)

{

less = travel[i, j];

choose[0] = travel[i, j - 3];

choose[1] = i;

choose[2] = (j - 3) / 4;

choose[3] = less;

}

}

}

}

}

return choose;

}

private int chk\_lv(int row\_of\_travel)//หาค่าระดับจากค่า row ในอาร์เรย์ travel

{

int lv = 5001;

lv = travel[row\_of\_travel - 1, 16] + 1;

return lv;

}

}

}

**Form2 (ทำการแสดงผลโดยการวาดตามลำดับในการค้นหา)**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace A\_Star\_Search

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

string[] city = new string[] { "Arad", "Bucharest", "Craiova", "Dobreta", "Eforie", "Fagaras", "Giurgiu", "Hirsova", "Iasi", "Lugoj",

"Mehadia", "Neamt", "Oradea", "Pitesti", "Rimnicu Vilcea", "Sibiu", "Timisoara", "Urziceni", "Vaslui", "Zerind"};

int[,] travel2 = new int[31, 17];//(เมือง,ระยะทางที่ห่างจากเมือง Bucharest,ระยะทางสะสม,ระยะทางรวม)X4 ,ระดับ

int[,] travel3 = new int[31, 17];//(เมือง,ระยะทางที่ห่างจากเมือง Bucharest,ระยะทางสะสม,ระยะทางรวม)X4 ,ระดับ แบบเรียงน้อยไปหามากในแต่ละ row

int[,] select\_city2 = new int[31, 4];//หมายเลขเมือง,ตำแหน่ง row ในอาร์เรย์ travel,ตำแหน่ง column ในอาร์เรย์ travel,ระยะทาง

Bitmap Bmp\_File = new Bitmap(1200, 650);

Bitmap vvv = new Bitmap(1200, 650);

int[,] point = new int[80, 4];//ตำแหน่งx,ตำแหน่งy,หมายเลขเมือง,ระยะทางรวม

int[] sch\_per\_lv;//เก็บค่าจำนวนครั้งในการค้นหาแต่ละระดับ

int[] c\_per\_lv;

int speed2, between\_y;

int time = 0, max\_lv = 0, sum\_step = 0, step = 0, c\_box = 0;

public Form2(int[,] travel, int[,] select\_city, int speed)

{

InitializeComponent();

travel2 = travel;

select\_city2 = select\_city;

speed2 = speed;

}

private void Form2\_Load(object sender, EventArgs e)

{

sch\_setup();

sort\_arr\_travel3();

///////////////////////////////////////////////////////////////////

between\_y = (650 - 120) / max\_lv;//หาระยะห่างของแต่ละระดับ

timer1.Interval = speed2;

timer1.Start();

}

private Bitmap drawbox(Bitmap Bmp, int loc\_x, int loc\_y, string st, int path, int sum, int sum\_path, Color color1)//วาดช่องมีขนาด กว้าง 90 สูง 40

{

Font font1 = new Font("Tahoma", 10.0f, FontStyle.Regular);

SolidBrush drawBrush = new SolidBrush(color1);

Graphics grap = Graphics.FromImage(Bmp);

Pen my\_pen = new Pen(color1, 1);

grap.DrawEllipse(my\_pen, loc\_x, loc\_y, 100, 60); // ----------------------------------->> กำหนดขนาดกรอบ grap.DrawEllipse(my\_pen, loc\_x, loc\_y, 90, 40);

//grap.DrawRectangle(my\_pen, loc\_x, loc\_y + 20, 90, 20);

grap.DrawString(" "+st, font1, drawBrush, loc\_x, loc\_y+10); // ----------------------------------->> กำหนดตำแหน่งข้อความ ชื่อเมือง

grap.DrawString(" "+sum\_path + "=" + sum + "+" + path, font1, drawBrush, loc\_x, loc\_y + 30); // ----------------------------------->> กำหนดตำแหน่งข้อความ ตัวเลข

return Bmp;

}

private Bitmap drawboxx(Bitmap Bmp, int loc\_x, int loc\_y,int locx, int locy, Color color1)

{

SolidBrush drawBrush = new SolidBrush(color1);

Graphics grap = Graphics.FromImage(Bmp);

Pen my\_pen = new Pen(color1, 1);

grap.DrawEllipse(my\_pen, loc\_x, loc\_y, 100, 60); // ----------------------------------->> กำหนดขนาดกรอบ grap.DrawEllipse(my\_pen, loc\_x, loc\_y, 90, 40);

grap.DrawEllipse(my\_pen, locx, locy, 100, 60);

return Bmp;

}

private Bitmap string\_end(Bitmap Bmp, int loc\_x, int loc\_y, string st)//วาดข้อความ Finished

{

Font font1 = new Font("Tahoma", 16.0f, FontStyle.Regular);

SolidBrush drawBrush = new SolidBrush(Color.DarkGreen);

Graphics grap = Graphics.FromImage(Bmp);

grap.DrawString(st, font1, drawBrush, loc\_x, loc\_y);

return Bmp;

}

private Bitmap draw\_line\_fn(Bitmap Bmp, int source\_x, int source\_y, int destination\_x, int destination\_y, int size, Color color2)//ลากเส้นจากตำแหน่ง source\_x,source\_y ไปยัง destination\_x,destination\_y

{

Point point1 = new Point(source\_x, source\_y+20);

Point point2 = new Point(destination\_x, destination\_y);

Pen my\_pen = new Pen(color2, size);

Graphics grap = Graphics.FromImage(Bmp);

grap.DrawLine(my\_pen, point1, point2);

return Bmp;

}

private int sch\_city\_per\_time(int stepp)//หาจำนวนเมืองที่สามารถไปได้ในขั้นตอนที่กำหนดมา

{

int count = 0;

for (int i = 3; i < 17; i += 4)

{

if (travel2[stepp, i] != 0)

{ count++; }

}

return count;

}

private int sch\_cbox\_from\_dis(int distance)//หาค่าว่าอยู่จุดที่เท่าไรจากระยะทางรวม

{

int box = 5000;

for (int i = 0; i < c\_box; i++)

{

if (point[i, 3] == distance)

{

box = i;

}

}

return box;

}

private int[] sort\_step(int step1)//เปรียบเทียบค่า4ค่าว่าอยู่ซ้ายถึงขวาโดยเลขน้อยอยู่ซ้าย

{

int[] order = new int[16];

int[] buf1 = new int[4];

int[] buf2 = new int[4];

for (int i = 0; i < 4; i++)

{

if (travel2[step1, i \* 4 + 3] != 0)

{

buf1[i] = travel2[step1, i \* 4 + 3];

buf2[i] = buf1[i];

}

}

Array.Sort(buf1);

for (int j = 0; j < 4; j++)

{

for (int k = 0; k < 4; k++)

{

if (buf2[j] == buf1[k])

{

order[k \* 4] = travel2[step1, j \* 4];

order[k \* 4 + 1] = travel2[step1, j \* 4 + 1];

order[k \* 4 + 2] = travel2[step1, j \* 4 + 2];

order[k \* 4 + 3] = travel2[step1, j \* 4 + 3];

}

}

}

return order;

}

private void sort\_arr\_travel3()//เรียงค่าน้อยสุดไปมากสุดแต่ละ row ในอาร์เรย์ travel3

{

int[] sort\_travel = new int[16];

int[] buf\_travel = new int[16];

for (int l = 0; l < sum\_step; l++)

{

Array.Clear(buf\_travel, 0, 16);

sort\_travel = sort\_step(l);

int index\_ = 0;

for (int m = 0; m < 4; m++)

{

if (sort\_travel[m \* 4 + 3] != 0)

{

index\_ = m \* 4;

m = 4;

}

}

Array.Copy(sort\_travel, index\_, buf\_travel, 0, buf\_travel.Length - index\_);

for (int n = 0; n < 16; n++)

{

travel3[l, n] = buf\_travel[n];

}

travel3[l, 16] = travel2[l, 16];

}

}

private void sch\_setup()//หาค่า ระดับสูงสุด,จำนวนครั้งในการค้นหา,จำนวนครั้งในการค้นหาแต่ละระดับ

{

for (int i = 0; i < 31; i++)

{

if (travel2[i, 16] > max\_lv)//หาว่ามีทั้งหมดถึงระดับที่เท่าไร นับเริ่มระดับที่ 0

{

max\_lv = travel2[i, 16];

}

if (travel2[i, 3] > 0) //หาว่าทำการท่องไปกี่ครั้งนับเริ่มครั้งที่ 0

{

sum\_step++;

}

}

sch\_per\_lv = new int[max\_lv + 1];//เก็บค่าจำนวนครั้งในการค้นหาแต่ละระดับ

c\_per\_lv = new int[max\_lv + 1];

for (int j = 0; j < max\_lv + 1; j++) // หาว่าแต่ละระดับมีการหาที่ระดับนี้กี่ครั้ง

{

for (int k = 0; k < sum\_step; k++)

{

if (travel2[k, 16] == j)

{

sch\_per\_lv[j]++;

}

}

}

}

private void timer1\_Tick(object sender, EventArgs e)

{

if (time == 1)

{

////////////วาดเมืองต้นทาง///////////

point[c\_box, 0] = 565;

point[c\_box, 1] = 10;

point[c\_box, 2] = travel3[0, 0];

point[c\_box, 3] = travel3[0, 3];

Bmp\_File = drawbox(Bmp\_File, point[c\_box, 0], point[c\_box, 1], city[point[c\_box, 2]], travel3[step, 1], travel3[step, 2], travel3[step, 3], Color.Red);

pictureBox1.Image = Bmp\_File;

c\_box++;

step++;

}

else if (time > 1)

{

if (time % 2 == 0)

{

if (sch\_per\_lv[travel3[step, 16]] == 1)

{

int city\_per\_time = sch\_city\_per\_time(step);

for (int i = 0; i < city\_per\_time; i++)

{

point[c\_box, 0] = 600 - city\_per\_time \* 45 + (i \* 110); // ------->> Default 45 + (i \* 90);

point[c\_box, 1] = 10 + between\_y \* travel3[step, 16];

point[c\_box, 2] = travel3[step, i \* 4];

point[c\_box, 3] = travel3[step, i \* 4 + 3];

Bmp\_File = drawbox(Bmp\_File, point[c\_box, 0], point[c\_box, 1], city[point[c\_box, 2]], travel3[step, i \* 4 + 1], travel3[step, i \* 4 + 2], travel3[step, i \* 4 + 3], Color.Black);//กรอบสีดำ

int box\_1 = sch\_cbox\_from\_dis(select\_city2[step - 1, 3]);

Bmp\_File = draw\_line\_fn(Bmp\_File, point[box\_1, 0] + 45, point[box\_1, 1] + 40, point[c\_box, 0] + 45, point[c\_box, 1], 2, Color.Black);//เส้นสีดำ

c\_box++;

}

pictureBox1.Image = Bmp\_File;

step++;

}

else if (sch\_per\_lv[travel3[step, 16]] == 2)

{

if (c\_per\_lv[travel3[step, 16]] == 0)

{

int city\_per\_time = sch\_city\_per\_time(step);

for (int i = 0; i < city\_per\_time; i++)

{

point[c\_box, 0] = 300 - city\_per\_time \* 45 + (i \* 110);

point[c\_box, 1] = 10 + between\_y \* travel3[step, 16];

point[c\_box, 2] = travel3[step, i \* 4];

point[c\_box, 3] = travel3[step, i \* 4 + 3];

Bmp\_File = drawbox(Bmp\_File, point[c\_box, 0], point[c\_box, 1], city[point[c\_box, 2]], travel3[step, i \* 4 + 1], travel3[step, i \* 4 + 2], travel3[step, i \* 4 + 3], Color.Black);

int box\_1 = sch\_cbox\_from\_dis(select\_city2[step - 1, 3]);

Bmp\_File = draw\_line\_fn(Bmp\_File, point[box\_1, 0] + 45, point[box\_1, 1] + 40, point[c\_box, 0] + 45, point[c\_box, 1], 2, Color.Black);

c\_box++;

}

pictureBox1.Image = Bmp\_File;

step++;

}

else if (c\_per\_lv[travel3[step, 16]] == 1)

{

int city\_per\_time = sch\_city\_per\_time(step);

for (int i = 0; i < city\_per\_time; i++)

{

point[c\_box, 0] = 900 - city\_per\_time \* 45 + (i \* 110);

point[c\_box, 1] = 10 + between\_y \* travel3[step, 16];

point[c\_box, 2] = travel3[step, i \* 4];

point[c\_box, 3] = travel3[step, i \* 4 + 3];

Bmp\_File = drawbox(Bmp\_File, point[c\_box, 0], point[c\_box, 1], city[point[c\_box, 2]], travel3[step, i \* 4 + 1], travel3[step, i \* 4 + 2], travel3[step, i \* 4 + 3], Color.Black);

int box\_1 = sch\_cbox\_from\_dis(select\_city2[step - 1, 3]);

Bmp\_File = draw\_line\_fn(Bmp\_File, point[box\_1, 0] + 45, point[box\_1, 1] + 40, point[c\_box, 0] + 45, point[c\_box, 1], 2, Color.Black);

c\_box++;

}

pictureBox1.Image = Bmp\_File;

step++;

}

}

else if (sch\_per\_lv[travel3[step, 16]] == 3)

{

if (c\_per\_lv[travel3[step, 16]] == 0)

{

int city\_per\_time = sch\_city\_per\_time(step);

for (int i = 0; i < city\_per\_time; i++)

{

point[c\_box, 0] = 200 - city\_per\_time \* 45 + (i \* 110);

point[c\_box, 1] = 10 + between\_y \* travel3[step, 16];

point[c\_box, 2] = travel3[step, i \* 4];

point[c\_box, 3] = travel3[step, i \* 4 + 3];

Bmp\_File = drawbox(Bmp\_File, point[c\_box, 0], point[c\_box, 1], city[point[c\_box, 2]], travel3[step, i \* 4 + 1], travel3[step, i \* 4 + 2], travel3[step, i \* 4 + 3], Color.Black);

int box\_1 = sch\_cbox\_from\_dis(select\_city2[step - 1, 3]);

Bmp\_File = draw\_line\_fn(Bmp\_File, point[box\_1, 0] + 45, point[box\_1, 1] + 40, point[c\_box, 0] + 45, point[c\_box, 1], 2, Color.Black);

c\_box++;

}

pictureBox1.Image = Bmp\_File;

step++;

}

else if (c\_per\_lv[travel3[step, 16]] == 1)

{

int city\_per\_time = sch\_city\_per\_time(step);

for (int i = 0; i < city\_per\_time; i++)

{

point[c\_box, 0] = 600 - city\_per\_time \* 45 + (i \* 110);

point[c\_box, 1] = 10 + between\_y \* travel3[step, 16];

point[c\_box, 2] = travel3[step, i \* 4];

point[c\_box, 3] = travel3[step, i \* 4 + 3];

Bmp\_File = drawbox(Bmp\_File, point[c\_box, 0], point[c\_box, 1], city[point[c\_box, 2]], travel3[step, i \* 4 + 1], travel3[step, i \* 4 + 2], travel3[step, i \* 4 + 3], Color.Black);

int box\_1 = sch\_cbox\_from\_dis(select\_city2[step - 1, 3]);

Bmp\_File = draw\_line\_fn(Bmp\_File, point[box\_1, 0] + 45, point[box\_1, 1] + 40, point[c\_box, 0] + 45, point[c\_box, 1], 2, Color.Black);

c\_box++;

}

pictureBox1.Image = Bmp\_File;

step++;

}

else if (c\_per\_lv[travel3[step, 16]] == 2)

{

int city\_per\_time = sch\_city\_per\_time(step);

for (int i = 0; i < city\_per\_time; i++)

{

point[c\_box, 0] = 1000 - city\_per\_time \* 45 + (i \* 110);

point[c\_box, 1] = 10 + between\_y \* travel3[step, 16];

point[c\_box, 2] = travel3[step, i \* 4];

point[c\_box, 3] = travel3[step, i \* 4 + 3];

Bmp\_File = drawbox(Bmp\_File, point[c\_box, 0], point[c\_box, 1], city[point[c\_box, 2]], travel3[step, i \* 4 + 1], travel3[step, i \* 4 + 2], travel3[step, i \* 4 + 3], Color.Black);

int box\_1 = sch\_cbox\_from\_dis(select\_city2[step - 1, 3]);

Bmp\_File = draw\_line\_fn(Bmp\_File, point[box\_1, 0] + 45, point[box\_1, 1] + 40, point[c\_box, 0] + 45, point[c\_box, 1], 2, Color.Black);

c\_box++;

}

pictureBox1.Image = Bmp\_File;

step++;

}

}

else if (sch\_per\_lv[travel3[step, 16]] == 4)

{

if (c\_per\_lv[travel3[step, 16]] == 0)

{

int city\_per\_time = sch\_city\_per\_time(step);

for (int i = 0; i < city\_per\_time; i++)

{

point[c\_box, 0] = 150 - city\_per\_time \* 45 + (i \* 110);

point[c\_box, 1] = 10 + between\_y \* travel3[step, 16];

point[c\_box, 2] = travel3[step, i \* 4];

point[c\_box, 3] = travel3[step, i \* 4 + 3];

Bmp\_File = drawbox(Bmp\_File, point[c\_box, 0], point[c\_box, 1], city[point[c\_box, 2]], travel3[step, i \* 4 + 1], travel3[step, i \* 4 + 2], travel3[step, i \* 4 + 3], Color.Black);

int box\_1 = sch\_cbox\_from\_dis(select\_city2[step - 1, 3]);

Bmp\_File = draw\_line\_fn(Bmp\_File, point[box\_1, 0] + 45, point[box\_1, 1] + 40, point[c\_box, 0] + 45, point[c\_box, 1], 2, Color.Black);

c\_box++;

}

pictureBox1.Image = Bmp\_File;

step++;

}

else if (c\_per\_lv[travel3[step, 16]] == 1)

{

int city\_per\_time = sch\_city\_per\_time(step);

for (int i = 0; i < city\_per\_time; i++)

{

point[c\_box, 0] = 450 - city\_per\_time \* 45 + (i \* 110);

point[c\_box, 1] = 10 + between\_y \* travel3[step, 16];

point[c\_box, 2] = travel3[step, i \* 4];

point[c\_box, 3] = travel3[step, i \* 4 + 3];

Bmp\_File = drawbox(Bmp\_File, point[c\_box, 0], point[c\_box, 1], city[point[c\_box, 2]], travel3[step, i \* 4 + 1], travel3[step, i \* 4 + 2], travel3[step, i \* 4 + 3], Color.Black);

int box\_1 = sch\_cbox\_from\_dis(select\_city2[step - 1, 3]);

Bmp\_File = draw\_line\_fn(Bmp\_File, point[box\_1, 0] + 45, point[box\_1, 1] + 40, point[c\_box, 0] + 45, point[c\_box, 1], 2, Color.Black);

c\_box++;

}

pictureBox1.Image = Bmp\_File;

step++;

}

else if (c\_per\_lv[travel3[step, 16]] == 2)

{

int city\_per\_time = sch\_city\_per\_time(step);

for (int i = 0; i < city\_per\_time; i++)

{

point[c\_box, 0] = 750 - city\_per\_time \* 45 + (i \* 110);

point[c\_box, 1] = 10 + between\_y \* travel3[step, 16];

point[c\_box, 2] = travel3[step, i \* 4];

point[c\_box, 3] = travel3[step, i \* 4 + 3];

Bmp\_File = drawbox(Bmp\_File, point[c\_box, 0], point[c\_box, 1], city[point[c\_box, 2]], travel3[step, i \* 4 + 1], travel3[step, i \* 4 + 2], travel3[step, i \* 4 + 3], Color.Black);

int box\_1 = sch\_cbox\_from\_dis(select\_city2[step - 1, 3]);

Bmp\_File = draw\_line\_fn(Bmp\_File, point[box\_1, 0] + 45, point[box\_1, 1] + 40, point[c\_box, 0] + 45, point[c\_box, 1], 2, Color.Black);

c\_box++;

}

pictureBox1.Image = Bmp\_File;

step++;

}

else if (c\_per\_lv[travel3[step, 16]] == 3)

{

int city\_per\_time = sch\_city\_per\_time(step);

for (int i = 0; i < city\_per\_time; i++)

{

point[c\_box, 0] = 1050 - city\_per\_time \* 45 + (i \* 110);

point[c\_box, 1] = 10 + between\_y \* travel3[step, 16];

point[c\_box, 2] = travel3[step, i \* 4];

point[c\_box, 3] = travel3[step, i \* 4 + 3];

Bmp\_File = drawbox(Bmp\_File, point[c\_box, 0], point[c\_box, 1], city[point[c\_box, 2]], travel3[step, i \* 4 + 1], travel3[step, i \* 4 + 2], travel3[step, i \* 4 + 3], Color.Black);

int box\_1 = sch\_cbox\_from\_dis(select\_city2[step - 1, 3]);

Bmp\_File = draw\_line\_fn(Bmp\_File, point[box\_1, 0] + 45, point[box\_1, 1] + 40, point[c\_box, 0] + 45, point[c\_box, 1], 2, Color.Black);

c\_box++;

}

pictureBox1.Image = Bmp\_File;

step++;

}

}

c\_per\_lv[travel3[step - 1, 16]]++;

}

if (time % 2 == 1)

{

////////ลากเส้นทางที่ทำการเลือก/////////

int box1 = sch\_cbox\_from\_dis(select\_city2[step - 2, 3]);

int box2 = sch\_cbox\_from\_dis(travel3[step - 1, 3]);

Bmp\_File = draw\_line\_fn(Bmp\_File, point[box1, 0] + 45, point[box1, 1] + 40, point[box2, 0] + 45, point[box2, 1], 4, Color.Blue);

Bmp\_File = drawboxx(Bmp\_File, point[box1, 0], point[box1, 1], point[box2, 0], point[box2, 1], Color.Red);

pictureBox1.Image = Bmp\_File;

}

}

if (time == sum\_step \* 2)

{

timer1.Stop();

pictureBox1.Image = string\_end(Bmp\_File, 550, 600, "Finished");

}

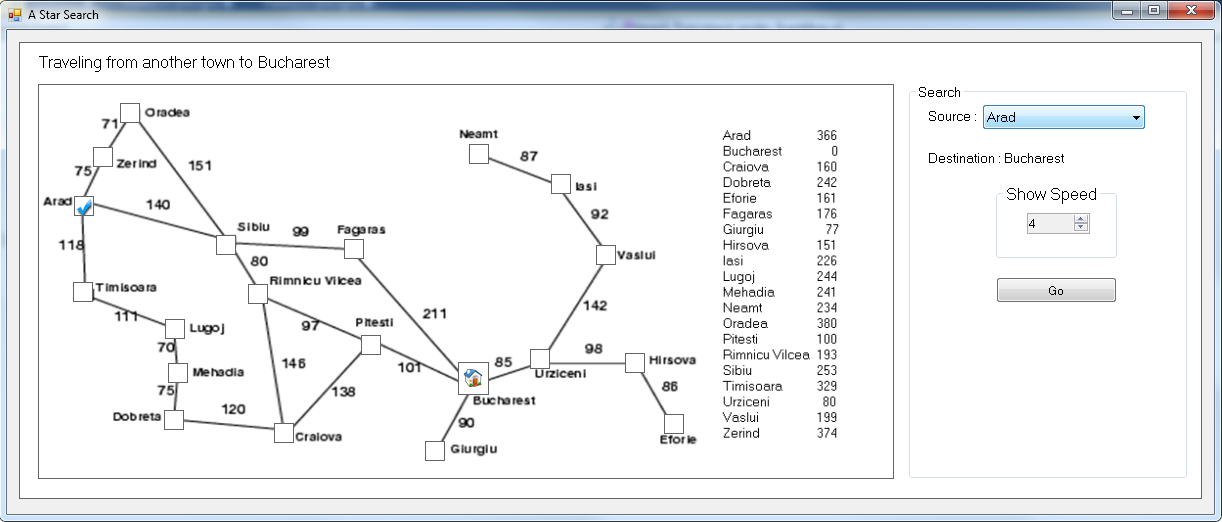
time++;

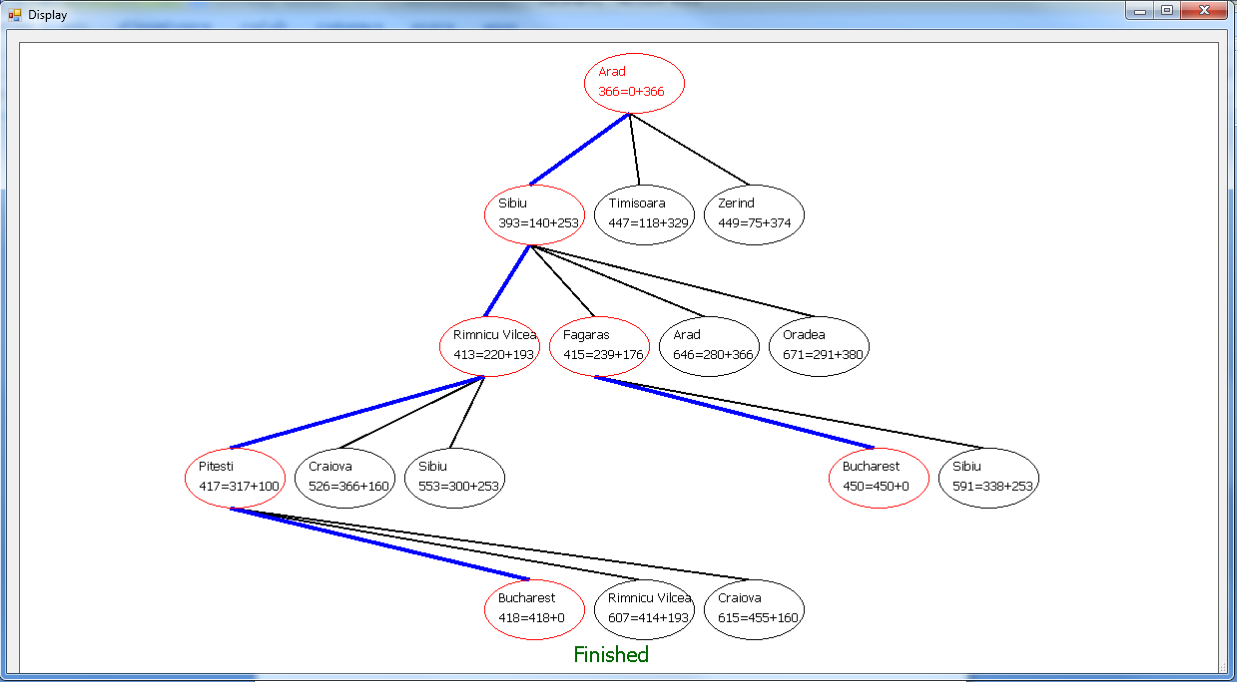
}

}

}

**ผลการรัน การค้นหาแบบ A \* Search**

**ยกตัวอย่างจากเมือง Arad**

**ค้นหาแบบ A \* Search เสร็จสิ้น**