Aufgabe5: Buhnenrennen

Lösungsidee:

Die Aufgabe erinnert sehr an ein Kürzester-Weg-Problem. Ist es für Maxi möglich einen Knoten vor Minni zu erreichen.
Der Plan der Buhnen und Löcher ist auch wie ein gerichteter Graph dargestellt und die Wege zwischen den Knoten sind dann die Gewichtungen der Kanten.
Zunächst müssen aus der Datei alle Buhnen eingelesen und jedes Loch von den Buhnen in einem Baum abgespeichert werden. Für jedes Loch gibt es einen Wert, der angibt wie lange der große Hund dorthin braucht und einen, der angibt wie lang der Weg des kleinen Hundes dorthin ist.
Da es nicht reicht, einfach nach dem kürzesten Weg für Minni zu suchen, da Maxi sie so auf dem Weg eventuell abfangen könnte, müssen die Buhnen Schritt für Schritt abgearbeitet werden.
Zunächst muss also der Weg des großen Hundes zu jedem Loch berechnet werden. Maxi passt zwar nicht durch die kleinen Löcher durch, könnte aber wenn er schnell genug ist Minni vor einem solchen abfangen. Deswegen muss die Zeit die Maxi zu jedem Loch braucht auch Schrittweise aufgebaut werden.
Es wird am Anfang von Maxis Startposition aus die erste Buhne mit allen Löchern(Minni- und Maxilöcher) genommen und mit dem Dijkstra-Algorithmus berechnet wie lange Maxi zu jedem Loch braucht. Dann wird diese Buhne überschrieben mit nur den Löchern von Maxi und anschließend wieder die nächste Buhne mit allen Löchern hinten angehängt. Dieser Vorgang muss solange wiederholt werden, bis alle Buhnen überprüft wurden.
Damit wurde die Zeit die Maxi zu jedem Knoten braucht berechnet.
Jetzt muss noch der Weg berechnet werden den Minni zu jedem Loch benötigt. Dafür wird wieder der Dijkstra-Algorithmus benutzt, aber diesmal muss dieser nur einmal angewandt werden, da Minni durch jedes Loch passt.
Anschließend muss nur noch Schrittweise ein Weg für Minni aufgebaut werden. Es wird von jeder Buhne der erste Knoten untersucht, wenn Minni diesen vor Maxi erreichen kann, wird er dem möglichen Fluchtweg für Minni angehangen und der gleiche Vorgang ein Buhne weiter wiederholt. Aber wenn Minni diesen Knoten nicht vor Maxi erreichen kann, wird dieser Knoten erst einmal als inaktiv angenommen, und der nächste Knoten von dieser Buhne untersucht(es wird Tiefen-Suche angewandt, damit eine Lösung möglichst schnell gefunden werden kann). Wenn es aber keinen Knoten mehr in dieser Buhne gibt, wird zu der vorherigen Buhne gewechselt, und der letzte Knoten von dem Fluchtplan entfernt.
Anschließend wird dieser Vorgang wiederholt bis entweder ein Fluchtplan für Minni gefunden wurde oder alle Möglichkeiten überprüft wurden.

Umsetzung:

Die Lösungsidee wird in C# umgesetzt.
Es werden Klassen für den Dijkstra-Algorithmus selber, die Kanten des Baumes und auch für die Blätter(Löcher) des Baumes benötigt.
Die Klasse für den Dijkstra-Algorithmus hat Methoden, um für eine Buhnenkonfiguration die kürzesten Wege zu rechnen, und eine, um die für einen Graphabschnitt benötigte Zeit zu berechnen.
Die Klasse für die Blätter des Baumes(Löcher) hat sieben Eigenschaften: ein eindeutiger Name, die X-Position, die Y-Position, ob er noch aktiv ist, die Weglänge für Minni, die Zeit die Maxi benötigt und noch den vorherigen Knoten im Baum.
Die Klasse für die Kanten des Baumes hat drei Eigenschaften: der Start, das Ende und die Distanz dieser Kante.
Bei dem Start des Programms wird die Datei mit den Buhnen eingelesen und eine 2-dimensionale Liste mit den zugehörigen Knoten für beide Hunde erstellt, und Knoten und Kanten erstellt. Daraufhin wird die Zeit, die Maxi zu jedem Knoten benötigt berechnet und bei denselben Knoten von Minnis Graph mit eingetragen.
Anschließend wird die doppelte Liste von Minni schrittweise durchlaufen, und jedes mal, wenn Minni vor Maxi ein Loch erreichen kann, wird die nächste Buhne untersucht.
Dies wird solange wiederholt, bis ein Fluchtplan erstellt wurde oder es keine Knoten mehr gibt, die noch aktiv sind.

Quelltext:

Die Funktion die mittels Dijkstra den kürzesten Weg zu jedem Knoten berechnet:

```
public void CalculateMinDistForNodes(List<Node> _nodes, List<List<Node>> _nodesLL,
                                      List<Edge> _edges, Node _start)
        {
            //die queue mit allen abzuarbeitenden nodes
            List<Node> list = new List<Node>();

            //die Liste sortieren
            _nodes.Sort((x, y) => x.cost.CompareTo(y.cost));

            //alle knoten der schlange hinzufuegen
            for (int i = 0; i < _nodes.Count; i++)
            {
                _nodes[i].cost = double.MaxValue;
                list.Add(_nodes[i]);
            }

            //die kosten fuer den startknoten gleich null setzten
            _start.cost = 0;

            //solange noch nodes in der queue sind
            while (list.Count != 0)
            {
                Node v = list[0];

                //alle Edges suchen...
                foreach (var e in _edges)
                {
                    //... die den Node am Anfang der Schlange als origin haben
                    if (e.Origin == v)
                    {
                        Node a = e.Destination;
                        Node b = e.Origin;

                        //wenn die kosten plus die kantenlaenge < als alte kosten
                        if (a.cost > b.cost + e.Distance)
                        {
                            //cost aktualisieren
                            a.cost = b.cost + e.Distance;
                            //vorherigen node setzten
                            a.previous = b;
                        }
                    }
                }

                //den node am anfang der Queue entfernen
                list.RemoveAt(0);

                //die Liste sortieren
                list.Sort((x, y) => x.cost.CompareTo(y.cost));

            }
        }
```

Die Funktion die die Zeiten zu jedem Knoten für Maxi berechnet:

```
for (int buhnen = 1; buhnen < nodesMinniLL.Count; buhnen++)
{
    maxiLL[buhnen - 1] = nodesMaxiLL[buhnen - 1];

    maxiLL.Add(nodesMinniLL[buhnen]);



    //die einfache Liste von Maxi
    maxi = new List<Node>();
    //alle loecher von der doppelteverketteten liste in eine einfache verschieben
    for (int x = 0; x < maxiLL.Count; x++)
    {
        for (int y = 0; y < maxiLL[x].Count; y++)
        {
            maxi.Add(maxiLL[x][y]);
        }
    }

    CreateEdges(maxiLL, edgesMaxi);
    //den kuerzesten Weg zu jedem Node berechnen
    dMaxi.CalculateMinDistForNodes(maxi, maxiLL, edgesMaxi, MaxiStart);

    for (int loecher = 0; loecher < nodesMinniLL[buhnen].Count; loecher++)
    {


        for (int names = 0; names < nodesMinniLL[buhnen].Count; names++)
        {
            if (maxiLL[buhnen][loecher].Name == nodesMinniLL[buhnen][loecher].Name)
            {
                nodesMinniLL[buhnen][loecher].maxiTime =
               dMaxi.TimeForRun(maxiLL[buhnen][loecher].cost, maxiVelocity);
                break;
            }
        }
    }
}
```

Beispiele:

In allen Beispielangaben gibt es eine Fluchtmöglichkeit für Minni und diese wird in wenigen Sekunden berechnet.

Buhnenrennen 1	Buhnenrennen 2	Buhnenrennen 3
M x - X = 70 || Y = 45,9 m - X = 140 || Y = 42,2 m - X = 210 || Y = 49,4	M x - X = 70 || Y = 14,1 m - X = 140 || Y = 28 m - X = 210 || Y = 67,9 m - X = 280 || Y = 64,5 x - X = 350 || Y = 56,55 m - X = 420 || Y = 50,3 m - X = 490 || Y = 97,85	M m - X = 70 || Y = 55,85 m - X = 140 || Y = 76,8 m - X = 210 || Y = 100,95 x - X = 280 || Y = 167,65 m - X = 350 || Y = 141,7 m - X = 420 || Y = 101,65 x - X = 490 || Y = 145,1
Buhnenrennen 4	**Buhnenrennen 5**	**Buhnenrennen 6**
M x - X = 70 || Y = 15,65 m - X = 140 || Y = 16,55 x - X = 210 || Y = 25,55 m - X = 280 || Y = 34,4 x - X = 350 || Y = 40,05 m - X = 420 || Y = 58,3 x - X = 490 || Y = 53,5	M x - X = 70 || Y = 28,25 m - X = 140 || Y = 63,2 x - X = 210 || Y = 42,05 m - X = 280 || Y = 34,95 m - X = 350 || Y = 12,85 m - X = 420 || Y = 28,1 x - X = 490 || Y = 27,5	M m - X = 70 || Y = 119,4 m - X = 140 || Y = 119,25 m - X = 210 || Y = 121,45 x - X = 280 || Y = 123,45 m - X = 350 || Y = 106,95 m - X = 420 || Y = 141,45 m - X = 490 || Y = 118,9 m - X = 560 || Y = 118,1 m - X = 630 || Y = 84,6 x - X = 700 || Y = 116,5 x - X = 770 || Y = 126,05 m - X = 840 || Y = 123 m - X = 910 || Y = 131,1 x - X = 980 || Y = 91,1 m - X = 1050 || Y = 71,55 m - X = 1120 || Y = 11,45 m - X = 1190 || Y = 26,1
Buhnenrennen 7	**Buhnenrennen 8**	**Buhnenrennen 9**
M m - X = 70 || Y = 173,5 m - X = 140 || Y = 149,4 x - X = 210 || Y = 191,9 m - X = 280 || Y = 187,7 m - X = 350 || Y = 256,5 m - X = 420 || Y = 269,55 m - X = 490 || Y = 175,25 x - X = 560 || Y = 184,5 x - X = 630 || Y = 152,3 x - X = 700 || Y = 91,4 m - X = 770 || Y = 71 m - X = 840 || Y = 111,75 m - X = 910 || Y = 137,95 x - X = 980 || Y = 131,55 x - X = 1050 || Y = 162,85 x - X = 1120 || Y = 173,95 m - X = 1190 || Y = 146,6 x - X = 1260 || Y = 120,5 m - X = 1330 || Y = 134,95 m - X = 1400 || Y = 176,75	M m - X = 70 || Y = 254 m - X = 140 || Y = 260 x - X = 210 || Y = 225,3 x - X = 280 || Y = 209 x - X = 350 || Y = 189,8 x - X = 420 || Y = 85,65 m - X = 490 || Y = 69,95 m - X = 560 || Y = 57 x - X = 630 || Y = 56,35 m - X = 700 || Y = 66,8 m - X = 770 || Y = 52,5 x - X = 840 || Y = 50,2 x - X = 910 || Y = 48,3 m - X = 980 || Y = 33,5 m - X = 1050 || Y = 114,3 x - X = 1120 || Y = 35,75 m - X = 1190 || Y = 126,55 m - X = 1260 || Y = 103,75 x - X = 1330 || Y = 129,55 m - X = 1400 || Y = 142,75 x - X = 1470 || Y = 121,75 m - X = 1540 || Y = 244,4 m - X = 1610 || Y = 175,6	M m - X = 70 || Y = 206,3 x - X = 140 || Y = 231,85 x - X = 210 || Y = 287,5 m - X = 280 || Y = 300,05 m - X = 350 || Y = 311,75 x - X = 420 || Y = 277,05 m - X = 490 || Y = 303,55 x - X = 560 || Y = 277,45 m - X = 630 || Y = 310,25 m - X = 700 || Y = 321,15 x - X = 770 || Y = 338,15 x - X = 840 || Y = 302,45 m - X = 910 || Y = 297,5 x - X = 980 || Y = 277,45 m - X = 1050 || Y = 243,25 x - X = 1120 || Y = 231,7 m - X = 1190 || Y = 225,65 m - X = 1260 || Y = 213,75 m - X = 1330 || Y = 186,4 x - X = 1400 || Y = 151,75 x - X = 1470 || Y = 192,5 x - X = 1540 || Y = 165,7 m - X = 1610 || Y = 192,95 m - X = 1680 || Y = 194,25 x - X = 1750 || Y = 285,65 m - X = 1820 || Y = 347,85
M m - X = 70 || Y = 287,65 x - X = 140 || Y = 317,4 m - X = 210 || Y = 318,15 x - X = 280 || Y = 268,25 m - X = 350 || Y = 206,5 m - X = 420 || Y = 165,25 x - X = 490 || Y = 282,25 m - X = 560 || Y = 270,75	M m - X = 70 || Y = 66,35 m - X = 140 || Y = 71,9 m - X = 210 || Y = 91,2 m - X = 280 || Y = 85,95 x - X = 350 || Y = 141,25 x - X = 420 || Y = 105,85 m - X = 490 || Y = 35,45 m - X = 560 || Y = 23,4	M x - X = 70 || Y = 158,45 m - X = 140 || Y = 178,4 m - X = 210 || Y = 173,15 x - X = 280 || Y = 383,5 x - X = 350 || Y = 437,6 m - X = 420 || Y = 327 m - X = 490 || Y = 197,5 m - X = 560 || Y = 155,8

x - X = 630 || Y = 247,15
m - X = 700 || Y = 151,05
x - X = 770 || Y = 264,65
m - X = 840 || Y = 248
m - X = 910 || Y = 166,45
x - X = 980 || Y = 34,2
m - X = 1050 || Y = 30,15
m - X = 1120 || Y = 64,4
m - X = 1190 || Y = 109,35
x - X = 1260 || Y = 79,2
m - X = 1330 || Y = 57,4
m - X = 1400 || Y = 157,85
m - X = 1470 || Y = 60,7
x - X = 1540 || Y = 79,25
x - X = 1610 || Y = 121,25
x - X = 1680 || Y = 121,8
m - X = 1750 || Y = 184,7
m - X = 1820 || Y = 186,85
m - X = 1890 || Y = 163,85
m - X = 1960 || Y = 112,3
x - X = 2030 || Y = 12,35

m - X = 630 || Y = 71,85
x - X = 700 || Y = 94,7
x - X = 770 || Y = 57,45
x - X = 840 || Y = 14,4
m - X = 910 || Y = 13,6
m - X = 980 || Y = 41,85
x - X = 1050 || Y = 64,9
x - X = 1120 || Y = 71,6
x - X = 1190 || Y = 51,7
m - X = 1260 || Y = 90
m - X = 1330 || Y = 192,6
m - X = 1400 || Y = 116,7
m - X = 1470 || Y = 130,75
m - X = 1540 || Y = 108,75
m - X = 1610 || Y = 70,85
m - X = 1680 || Y = 109,15
m - X = 1750 || Y = 78,55
x - X = 1820 || Y = 43,8
m - X = 1890 || Y = 53,95
x - X = 1960 || Y = 46,15
m - X = 2030 || Y = 54,65
m - X = 2100 || Y = 120,7
x - X = 2170 || Y = 128,85
x - X = 2240 || Y = 101,1

m - X = 630 || Y = 81,55
x - X = 700 || Y = 160,2
x - X = 770 || Y = 93,25
x - X = 840 || Y = 94,2
x - X = 910 || Y = 71
m - X = 980 || Y = 61,05
x - X = 1050 || Y = 12,95
x - X = 1120 || Y = 21,15
m - X = 1190 || Y = 19,1
m - X = 1260 || Y = 75,15
m - X = 1330 || Y = 90,5
m - X = 1400 || Y = 129,7
x - X = 1470 || Y = 257,45
x - X = 1540 || Y = 294,95
x - X = 1610 || Y = 198,6
m - X = 1680 || Y = 285,2
m - X = 1750 || Y = 261,4
m - X = 1820 || Y = 206,6
x - X = 1890 || Y = 213,85
x - X = 1960 || Y = 150,5
m - X = 2030 || Y = 116,65
x - X = 2100 || Y = 99,95
x - X = 2170 || Y = 108,8
m - X = 2240 || Y = 186,6
x - X = 2310 || Y = 183,8
x - X = 2380 || Y = 200,2
m - X = 2450 || Y = 249,3

M
m - X = 70 || Y = 55,4
m - X = 140 || Y = 83,9
x - X = 210 || Y = 12,75
m - X = 280 || Y = 133,65
m - X = 350 || Y = 105,2
m - X = 420 || Y = 174,85
m - X = 490 || Y = 358,85
m - X = 560 || Y = 414,35
m - X = 630 || Y = 449,45
m - X = 700 || Y = 427,75
m - X = 770 || Y = 440
m - X = 840 || Y = 442,7
x - X = 910 || Y = 455,4
m - X = 980 || Y = 415,3
m - X = 1050 || Y = 395
m - X = 1120 || Y = 391,4
x - X = 1190 || Y = 422
x - X = 1260 || Y = 431,5
x - X = 1330 || Y = 415,85
m - X = 1400 || Y = 469,45
x - X = 1470 || Y = 462
m - X = 1540 || Y = 443,75
x - X = 1610 || Y = 461,15
m - X = 1680 || Y = 473,2
m - X = 1750 || Y = 328
x - X = 1820 || Y = 380,35
m - X = 1890 || Y = 369,15
m - X = 1960 || Y = 257,1
x - X = 2030 || Y = 241,55
x - X = 2100 || Y = 233
m - X = 2170 || Y = 349,65
m - X = 2240 || Y = 338,25
m - X = 2310 || Y = 347,35
x - X = 2380 || Y = 382,6
x - X = 2450 || Y = 335,15
m - X = 2520 || Y = 349,05
m - X = 2590 || Y = 339
m - X = 2660 || Y = 266,75

M
m - X = 70 || Y = 217,6
x - X = 140 || Y = 128,85
m - X = 210 || Y = 12,6
m - X = 280 || Y = 44,6
m - X = 350 || Y = 137,05
x - X = 420 || Y = 458,05
m - X = 490 || Y = 438,15
x - X = 560 || Y = 361,05
m - X = 630 || Y = 386,8
x - X = 700 || Y = 417,7
m - X = 770 || Y = 476,7
x - X = 840 || Y = 451,85
x - X = 910 || Y = 404,55
m - X = 980 || Y = 437
x - X = 1050 || Y = 447,35
m - X = 1120 || Y = 415,4
m - X = 1190 || Y = 405,9
x - X = 1260 || Y = 400,75
x - X = 1330 || Y = 416,6
m - X = 1400 || Y = 380,6
m - X = 1470 || Y = 227,75
x - X = 1540 || Y = 154,75
m - X = 1610 || Y = 176,85
m - X = 1680 || Y = 173,65
x - X = 1750 || Y = 151,1
m - X = 1820 || Y = 131,75
m - X = 1890 || Y = 147,45
x - X = 1960 || Y = 280,25
x - X = 2030 || Y = 226,4
m - X = 2100 || Y = 236,15
x - X = 2170 || Y = 236,65
m - X = 2240 || Y = 141,6
m - X = 2310 || Y = 104,15
m - X = 2380 || Y = 102,85
m - X = 2450 || Y = 78,6
m - X = 2520 || Y = 105,8
x - X = 2590 || Y = 30,85
m - X = 2660 || Y = 92,5
x - X = 2730 || Y = 107,4
m - X = 2800 || Y = 86
m - X = 2870 || Y = 39,55