KOCAELİ ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ

Yazılım Geliştirme Laboratuvarı II

GEZGİN SATICI PROBLEMİ

BATUHAN RAPATA 181307005

BERK AKIN 191307012

RECEP TAYYİP ÇAKIR 191307002

ONUR AKYILDIZ 191307026

KOCAELİ 2022

ÖZET

Bir programda düğümler arasında en kısa yolu bulan algoritmaya Dijkstra algoritması denir. 1956 yılında Edsger W. Dijkstra tarafından ortaya çıkmıştır. Günümüzde haritalar uygulamasında gideceğiniz lokasyonda en kısa yolları bulan algoritmadır. Dijkstra algoritmasının yanı sıra graph algoritmasınıda kullandık düğümler arası mesafeyi directed weighted graph şeklinde gösterip dijkstra algoritmasıyla en kısa mesafeyi buldurduk.

Gezgin satıcı uygulmasında djikstra algoritmasını kullanmamızın sebebide budur, örneğin günümüzde revaçta olan getir uygulamasında kurye alıcıya en kısa sürede ürünü götürmesi için en kısa yolu bilmesi gereklidir.

ABSTRACT

The algorithm that finds the shortest path between nodes in a program is called Dijkstra's algorithm. It was developed by Edsger W. Dijkstra in 1956. For example nowadays it is the algorithm which is find the shortest routes at the location you will go to in the map application. We used the graph algorithm as well as the Dijkstra algorithm, we showed the distance between the nodes in a directed weighted graph and found the shortest distance with the dijkstra algorithm.

This is the reason why we use the djikstra algorithm in the traveling seller application, for example, in today's popular application “Getir”, the courier needs to know the shortest route to deliver the product to the buyer as soon as possible.

Keywords: Graph, Desktop Application, Dijsktra, nodes, Path

GİRİŞ

Günümüzde özellikle Covid zamanında herkes online alışverişe yöneldi, işletmelere gezgin satıcı programı ihtiyacı duyuldu bu sebebtendirki getir, migros sanal market, istegelsin gibi uygulamalar geliştirilidi ve hepsinde dijsktra algoritması kullanıldı.

Dijkstra algoritması düğümlar arasındaki mesafeleri ele alarak ulaşılmak istenilen düğüme en kısa yolu bulan algoritmadır. Bu sebeple dijsktra algoritması kullanmamız gerekliydi, yanı sıra graph algoritmasını da düğümlar arası mesafeyi göstermek için kullandık.

1. PROBLEMİN TANIMI

Gezgin satıcı problemini düşünürsek, her dağıtımda etrafındaki insanlara sorarak konum bulamazsın, müşteri memnuniyeti azalır bu sebeptendirki en kısa yolu bulan bi programa ihtiyaç vardır.

2. YAPILAN ARAŞTIRMALAR

Projemizin yapımına başlamadan önce dijsktra algoritması hakkında araştırmalar yaptık, makaleler okuduk. Bu algoritmanın işleyişini, oluşturan unsurları ve bunların birbirleriyle bağlantılarını belirledik. Ardından bulduğumuz bilgiler yardımıyla projemizin ara yüzünü oluşturduk. Bu adımdan sonra kod yazma işlemine geçtiğimizde oluşturduğumuz arayüze bağlı kalarak graph ve dijsktra algoritmalarını ekledik. Zamanla arayüzde değişiklik yaparak göze hoş hitap etmesini sağladık. Ardından projemizin veritabanı kısmını yapmaya başladık. İlk aşamada kağıt üzerinde nası bi veritabanı daha kullanışlı olur diye grupça tartıştık. ER diyagramını çıkarıp SQL kodların yazmaya başladık. Bu aşamadan sonra veri tabanı ve program arasındaki bağı oluşturmaya çalışırken SQL sorguları ve kullandığımız komutlar çok sayıda error verdi. StackOverflow ve GitHub gibi internet sitelerinden yardım alarak hatalarımızı çözdük.

```
private int TempVertexMinPL()
{
    int min = INFINITY;
    int x = NIL;
    for (int v = 0; v < n; v++)
    {
        if (vertexList[v].status == TEMPORARY && vertexList[v].pathLength < min)
        {
            min = vertexList[v].pathLength;
            x = v;
        }
    }
    return x;
}

public void FindPaths(String source)
{
    int s = GetIndex(source);

    Dijkstra(s);

    Console.WriteLine("kaynak : " + source + "\n");

    for (int v = 0; v < n; v++)
    {
        Console.WriteLine("vars : " + vertexList[v].name);
        if (vertexList[v].pathLength == INFINITY)
            Console.WriteLine("There is no path from " + source + " to vertex " + vertexList[v].name + "\n");
        else
            FindPath(s, v);
    }

}

private void FindPath(int s, int v)
{
    int i, u;
    int[] path = new int[n];
    int sd = 0;
    int count = 0;

    while (v != s)
    {
        count++;
        path[count] = v;
        u = vertexList[v].predecessor;
```

Şekil 1: weighted directed graph

```
int count = 0;
count = latitute.Count;//kaç adet koordinat olacak
string[] coordinates = new string[count];//koordinat sayısı kadar büyüklüğünde string dizisi
DirectedWeightedGraph g = new DirectedWeightedGraph();//directed graph constructor class


foreach (var item in etiket)//konumların etiketleri ile node oluşturmak
{
    g.InsertVertex(item);
}

//daha sonra tüm koordinatlar arası uzaklığı bulmak için distances metoduna gönderilecek konumların dizisye aktarılması
for (int i = 0; i < latitute.Count; i++)
{

    richTextBox1.AppendText(latitute[i] + "," + longtitute[i] + "\n");//koordinatların yazılış şeması richtextbox
    coordinates[i] = latitute[i].Substring(0, 8) + "," + longtitute[i].Substring(0, 8);//koordinat dizisine gönderilen koordinatlar
}


for (int i = 0; i < latitute.Count; i++)//her node arasındaki mesafelerin apiye gönderilerek bulunması
{

    baslangic = latitute[i].Substring(0, 8) + "," + longtitute[i].Substring(0, 8);
    List<int> km = new List<int>();//return edilen nodelar arası mesafenin tutulduğu list
    km = Distances(baslangic, coordinates, etiket);
    richTextBox1.AppendText("\n\nKOORDINAT: " + coordinates[i]);
    for (int j = 0; j < etiket.Count; j++)
    {
        if (km[j] != 0)
        {
            g.InsertEdge(etiket[i], etiket[j], Convert.ToInt32(km[j]));
            richTextBox1.AppendText("\nBAŞLANGIÇ: " + etiket[i] + " VARIŞ: " + etiket[j] + " MESAFE: " + km[j]);
        }
    }
}
for (int i = 0; i < etiket.Count; i++)//sırayla tüm nodeler başlangıç olarak seçilir ve diğer noktalara en kısa yolu bulur
{
    g.FindPaths(etiket[i]);
}
```

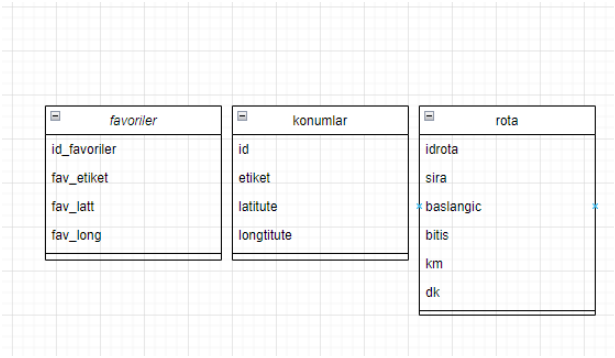
Şekil 2: dijsktra algoritması

3. ARKA PLAN KODLARI

Graph ve dijkstra araştırmasından sonra isterlere bağlı kalarak circkle.cs adlı dosyada daireler oluşturup eklediğimiz google maps üzerine başlangıç noktası olarak ekledik. Veri yapıları dersinden kalan bilgiler ve StackOverflow gibi yardımlaşma sitelerinden yardım alarak dijsktra algoritmasını başarılı bir şekilde eklemesini yaptık. DirectedWeightedGraph.cs adlı dosyada ise düğümler arası mesafeyi bi grapha gönderip gerekli hesaplamaları yaptık. Form1.cs adlı dosyada ise gerekli olan dijsktra algoritmasını ekledik graphtan aldığı bilgilerle en kısa yolu buldurduk. Kodları yorum satırlarıyla anlaşılır şekle getirdik. Google haritaları eklemek için GMap adında eklentiyi kullandık web request oluşturup kişiye özel token sayesinde google haritaları ekledik. Rect.cs dosyasında konumların varış noktalarını kare şeklinde gösterdik

4. YAZILIM MİMARİSİ

Programımıza en uygun programlama dilinin C# olduğuna karar verdik. Daha sonra bu dili kullanarak Microsoft Visual Studio 2019 IDE'sinde projemizi gerçekleştirdik. Veri depolama olarak ise MySQL Veritabanı Yönetim Sistemi'ni kullanmaya karar verdik ve projemizi gerçekledik.



Şekil 3: ER diyagramı

5. GENEL YAPI

Gezgin satıcı programımız bir işletmenin ihtiyacı olan her türlü bilgiyi işleme ve düzenlemeye yarayacak şekilde yapıldı. Düğümler arası mesafeyi, düğümlerin konumları, google haritaları düzgün şekilde göstermeye kadar her türlü veri kullanıcının rahatça ulaşabileceği ve düzenleyebileceği şekilde dizayn edildi.

Ara yüz tasarımımızın kullanıcı dostu olmasına özen gösterdik. Karmaşadan uzak, öğrenilmesi ve kullanılması kolay bir dizayna sahip olması için çalıştık. Seçtiğimiz ikonlar yardımıyla kullanıcıyı sıkmayacak bir görünüm kazandırdık. Göz yormayacak renkler ile de hoş bir görünüm elde etmesini sağladık.

6. KAYNAKÇA

[1] https://stackoverflow.com/questions/8136384/maps-with-windows-forms-application

[2] https://stackoverflow.com/questions/12774001/implementing-dijkstras-algorithm-into-windows-form-chart-control

[3] https://www.geeksforgeeks.org/shortest-path-weighted-graph-weight-edge-1-2/

[4] https://www.youtube.com/watch?v=n7Ojngjx0B8&list=PLlD7n_T-mUjVuqIhWVfaNhnpqCZmNcA9e&index=2&t=4s

[5] https://www.youtube.com/watch?v=eXJ1qkTMLw8&list=PLlD7n_T-mUjVuqIhWVfaNhnpqCZmNcA9e&index=3

[6] https://www.youtube.com/watch?v=TxSJJfaAzKg&list=PLlD7n_T-mUjVuqIhWVfaNhnpqCZmNcA9e&index=4&t=9s

[7] https://www.youtube.com/watch?v=WpfjRaYVId8&list=PLlD7n_T-mUjVuqIhWVfaNhnpqCZmNcA9e&index=5

[8] https://www.youtube.com/watch?v=hM7ZQwx4YFI&list=PLlD7n_T-mUjVuqIhWVfaNhnpqCZmNcA9e&index=6

[9] https://www.youtube.com/watch?v=iDzRLRn0U9k&list=PLlD7n_T-mUjVuqIhWVfaNhnpqCZmNcA9e&index=7

[10] https://www.youtube.com/watch?v=FF-PJQxpjOY&list=PLlD7n_T-mUjVuqIhWVfaNhnpqCZmNcA9e&index=8&t=720s

[11] https://www.youtube.com/watch?v=istT6E9Mnds&list=PLlD7n_T-mUjVuqIhWVfaNhnpqCZmNcA9e&index=9

[12] https://www.youtube.com/watch?v=KP_48J-WPg8&list=PLlD7n_T-mUjVuqIhWVfaNhnpqCZmNcA9e&index=12

[13] https://www.youtube.com/watch?v=BLWXVqqrAa4&list=PLlD7n_T-mUjVuqIhWVfaNhnpqCZmNcA9e&index=11

[14] https://www.youtube.com/watch?v=ZyyU6MfbXvc&list=PLlD7n_T-mUjVuqIhWVfaNhnpqCZmNcA9e&index=10

[15] https://www.youtube.com/watch?v=Z9PFVZDp4bM&list=PLlD7n_T-mUjVuqIhWVfaNhnpqCZmNcA9e&index=13

[16] https://www.youtube.com/watch?v=zlDyYm_7MJg&list=PLlD7n_T-mUjVuqIhWVfaNhnpqCZmNcA9e&index=14

[17] https://www.youtube.com/watch?v=61uZG2wddF8&list=PLlD7n_T-mUjVuqIhWVfaNhnpqCZmNcA9e&index=15

[18] https://www.youtube.com/watch?v=WifWNXLT0ng&list=PLlD7n_T-mUjVuqIhWVfaNhnpqCZmNcA9e&index=17

[19] https://www.youtube.com/watch?v=07ypcnNhZ-k&list=PLlD7n_T-mUjVuqIhWVfaNhnpqCZmNcA9e&index=18

[20] https://www.youtube.com/watch?v=aneOwy1sP98&list=PLlD7n_T-mUjVuqIhWVfaNhnpqCZmNcA9e&index=19

[21] https://www.youtube.com/watch?v=3HkeJmNaxWM&list=PLlD7n_T-mUjVuqIhWVfaNhnpqCZmNcA9e&index=20
[22] https://www.youtube.com/watch?v=_J_wLIIh7jk&list=PLlD7n_T-mUjVuqIhWVfaNhnpqCZmNcA9e&index=21
[23] https://stackoverflow.com/questions/1025670/how-do-you-automatically-resize-columns-in-a-datagridview-control-and-allow-the
[24] https://maps.googleapis.com/maps/api/directions/json?origin=
[25] https://github.com/radioman/greatmaps/tree/master/Demo.WindowsForms/CustomMarkers
[26] https://github.com/radioman/greatmaps/tree/master/Demo.WindowsForms/CustomMarkers
[27] https://www.youtube.com/watch?v=KmVSCv6Bn8E
[28] https://gist.github.com/ashwath10110/5e0bc687f7a56ea43284