

GEZGİN KARGO UYGULAMASI

Tuba Salgın, Mustafa Yiğit

180201095-180201108

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

tubaslgn99@icloud.com , mustafaahmet_60@hotmail.com

1.Özet:

Gezgin Kargo Uygulaması kullanıcının girdiği şehirlere göre en kısa yolu bulmak için yapılmış bir programdır. Bunu yaparken dijkstra algoritması kullanılır. Komşuluk matrisi kullanılarak en kısa yol bulunur.

2.Temel Bilgiler:

Proje Java programlama dilinde geliştirilmiş olup IDE olarak IntelliJ IDEA kullanılmıştır.

3.Class:

3.1.Graph.java:

Graph sınıfı 81 adet şehrin birbirine komşuluklarına göre bağlanmasını sağlar.

3.1.1:addEdge fonk.:

Bu fonksiyon grafın elemanlarını eklemeyi yapar.

3.1.2:printMatrix fonk.:

Dosyadan okunan bilgilerin matris şeklinde yazılmasını sağlar. Eğer dosyadan şehrin komşuluğuna ulaşmışsa ulaştığı değeri yazdırır. Komşusu değilse “|” işaretini basar.

3.1.3:printEdges fonk.:

Tüm illerin komşu plaka kodlarını ekrana yazmak için kullanılmıştır.

3.2.EnKisaYol.java:

Bu sınıf dijkstra algoritmasının kullanılarak en kısa yolu bulmak için kullanılan fonksiyonları içerir. Bu algoritma kaynakçada belirtilen sitelerden yardım alınarak oluşturulmuştur.

3.2.1:dijkstra fonk.:

Dijkstra() fonksiyonu 3 parametre(graf, başlangıç, bitiş) alan ve minimum mesafeyi bulan fonksiyondur.

3.2.3:printSolution fonk.:

Dijkstra() fonksiyonun içinde yine EnKisaYol sınıfında tanımlanmış ve mesafeyi yazdırmak için kullanılmıştır. İçine parametre olarak dijkstra fonksiyonunda tanımlanmış 81 boyutluk, içinde mesafelerin olduğu dizi ve hangi şehre gidileceği bilgisini tutan parametreler gönderilmiştir.

3.3.Değişkenler.java:

Bu sınıf EnKisaYol sınıfında, printSolution() fonksiyonun içinde toplam mesafeyi bulmak için kullanılmıştır.

3.4.GUI.java:

Main sınıfının içinde işlemlerin yapılması yerine tüm işlemler bu sınıfta yapılmıştır. Swing yapısı kullanılarak tasarım oluşturulmuştur. Ayrıca dosya okunma işlemleri de bu sınıfta oluşturulmuştur.

3.5.grafik.java:

Haritanın üzerinde gidilen şehirleri yani yolu çizdirmek için kullanılmıştır. Çizimi yapmak için drawLine() fonksiyonu kullanılmıştır.

3.6.main.java:

Yukarıda da yazıldığı gibi main içindeki işlemler GUI sınıfına taşınmıştır. Main içinde sadece GUI sınıfından nesne üretilmiştir.

4.Karşılaşılan Problemler:

Dijkstra algoritması oluşturulurken birçok yöntem denenmiştir. Fakat başarısız olduğu için kaynakçada belirtilen sitelerden yardım alınmıştır.

Proje istenildiği gibi en kısa mesafenin ne kadar olacağını komşuluk matrisini kullanarak doğru bir şekilde hesaplamaktadır, fakat geçtiği illeri ekrana yazdıramamaktadır.

6.Sonuç:

Karşılaşılan problemlerde verildiği gibi geçilen şehirlerin yazdırılması haricinde proje istenilenleri yapmaktadır.

7.Projede Kullanılan Dosyalar:

Projede şehirlerin komşulukları ve komşu olan şehirlerin birbirine olan uzaklıklarının tutulduğu bir txt uzantılı dosya, kullanıcının gideceği şehirleri seçtiği ekranda hangi plakanın hangi şehre ait olduğunu görebilmesi için bir plakaların bulunduğu fotoğraf, haritanın görsel bir şekilde ekrana gelmesi ve yolun gösterilmesi için Türkiye haritası fotoğrafı ve son olarak bu fotoğraf da çizim yapabilmek için fotoğrafın koordinatlarını tutan HARITAKoordinat adında bir metin belgesi bulunmaktadır.

7.1. Verilen komsuuzaklik.txt dosyasının okunması:

```
> main
```

↑ ↓ 	1,Adana,0,69,0,0,0,0,335,0,0,0,0,0,0,192,0,0, 2,0,Adiyaman,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,207,0,0,0,0,158,0,0,0,0,0,0,0,0,187,0,163, 3,0,Afyonkarahisar,0,0,0,0,0,0,0,0,0,0,0,169,0,0,0,0,222,0,0,0,0,144,0,0,0,0,168,0,0,0,0,0,0,223,1 4,0,0,Agrı,0,0,0,0,0,0,0,234,0,0,0,0,0,0,0,0,0,183,0,0,0,0,0,0,214,0,0,0,0,0,0,0,0,2 5,0,0,0,Amasya,0,0,0,0,0,0,0,0,0,0,0,92,0,2 6,0,0,0,Ankara,0,0,0,0,0,0,0,191,0,0,0,138,0,0,0,0,0,233,0,0,0,0,0,0,0,0,0,0,0,184,0,258,0,0, 7,0,0,0,0,Antalya,0,0,0,0,0,0,0,122,0,0,0,0,0,0,0,0,0,0,0,0,138,466,0,0,0,0,0,0,322,0,0,0,0, 8,0,0,0,0,0,Arztvin,0,0,0,0,0,0,0,0,0,0,0,0,0,0,226,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 9,0,0,0,0,0,Erdinç,0,0,0,0,0,0,0,0,0,126,0,0,0,0,0,0,0,0,0,0,0,0,126,0,0,0,0,0,0,0,155,0,99, 10,0,0,0,0,0,0,Balıkesir,0,0,0,0,0,151,198,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,176,0,0,0,0,0,0,228,0,141 11,0,0,0,0,0,0,0,Bilecik,0,0,213,0,94,0,112,0,0,0, 12,0,0,0,0,0,0,0,Bingöl,0,0,0,0,0,0,0,0,0,141,0,144,275,177,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 13,0,0,0,234,0,0,0,0,0,0,0,Bitlis,0,83, 14,0,0,0,0,0,191,0,0,0,0,213,0,0,0,0,0,233,0,0,0,0,0,0,291,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 15,0,169,0,0,0,122,0,0,0,0,0,0,0,Burdur,0,0,0,0,158,0,0,0,0,0,0,0,0,0,0,51,0,0,0,0,0,0,0,0,0,0,2 16,0,0,0,0,0,0,0,151,94,0,0,0,Bursa,0,132,0,177,0,0,0, 17,0,0,0,0,0,0,0,0,198,0,0,0,0,0,Canakkale,0,0,0,216,0, 18,0,0,0,0,0,130,0,0,0,0,0,233,0,0,0,Cankiri,156,0,0,0,0,0,0,0,0,0,0,0,0,0,0,106,0,0,0,0,0,0, 19,0,0,0,92,0,0,0,0,0,0,0,0,0,0,156,Corum,0,0,0,0,0,0,0,0,0,0,0,0,0,0,197,0,0,0,0,0,0,0,0,0, 20,0,222,0,0,0,0,0,126,0,0,0,0,158,0,0,0,Denizli,0,288,0 21,0,207,0,0,0,0,0,0,0,141,0,0,0,0,0,0,0,Diyanbakır,0,153,0,0,0,0,0,0,0,0,0,0,0,0,0,0,249,0 22,0,0,0,0,0,0,0,0,0,0,0,0,216,0,0,0,Edirne,0,0,0,0,0,0,0,0,0,0,0,0,0,0,62,0,0,0,0,0,0,0,0,23 23,0,0,0,0,0,0,0,0,0,0,144,0,0,0,0,0,0,0,Elazığ,267,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,98,0,0,0,
----------------	---

TODOLIST

Run Messages Terminal

The screenshot shows an IDE with a C++ program and its output. The program uses a 2D array to store numbers. The output window displays the numbers in a grid-like pattern, with some numbers appearing multiple times.

Program Code:

```

105     f++;
106     }
107 }

```

Output:

```

151.0
94.0
213.0
169.0
234.0
191.0
122.0
213.0
151.0
198.0
94.0
233.0
150.0
287.0
222.0
92.0
126.0
141.0
144.0
275.0
177.0
191.0
150.0
144.0
183.0
233.0
226.0
83.0
291.0
168.0
130.0
51.0

```

```

1 plakalı sehrin komsu il plakaları: 31 33 38 46 51 80
2 plakalı sehrin komsu il plakaları: 21 27 44 46 63
3 plakalı sehrin komsu il plakaları: 15 20 26 32 42 43 64
4 plakalı sehrin komsu il plakaları: 13 25 36 49 65 76
5 plakalı sehrin komsu il plakaları: 19 55 60 66
6 plakalı sehrin komsu il plakaları: 14 18 26 40 42 68 71
7 plakalı sehrin komsu il plakaları: 15 32 33 42 48 70
8 plakalı sehrin komsu il plakaları: 25 53 75
9 plakalı sehrin komsu il plakaları: 20 35 45 48
10 plakalı sehrin komsu il plakaları: 16 17 35 43 45
11 plakalı sehrin komsu il plakaları: 14 16 26 43 54
12 plakalı sehrin komsu il plakaları: 21 23 24 25 49 62
13 plakalı sehrin komsu il plakaları: 4 49 56 65 72
14 plakalı sehrin komsu il plakaları: 6 11 18 26 54 67 78 81
15 plakalı sehrin komsu il plakaları: 3 7 20 32 48
16 plakalı sehrin komsu il plakaları: 10 11 41 43 54 77
17 plakalı sehrin komsu il plakaları: 10 22 59
18 plakalı sehrin komsu il plakaları: 6 14 19 37 71 78
19 plakalı sehrin komsu il plakaları: 5 18 37 55 57 66 71
20 plakalı sehrin komsu il plakaları: 3 9 15 45 48 64 71
21 plakalı sehrin komsu il plakaları: 2 12 23 44 47 49 63 72
22 plakalı sehrin komsu il plakaları: 17 39 59
23 plakalı sehrin komsu il plakaları: 12 21 24 44 62

```

Build completed successfully in 1 s 678 ms (6 minutes ago)

7.4. Gidilecek şehirlerin seçilmesi için kullanıcıya sunulan ekran:

HARITA

Lütfen aşağıda bulunan şehirlerden eklemek istediğinizi seçip, ekle butonuna basın.

UYRAI : Başlangıç noktası ve bitiş noktası Kocaeli olarak belirlenmiştir. Girmenize gerek yok!

47

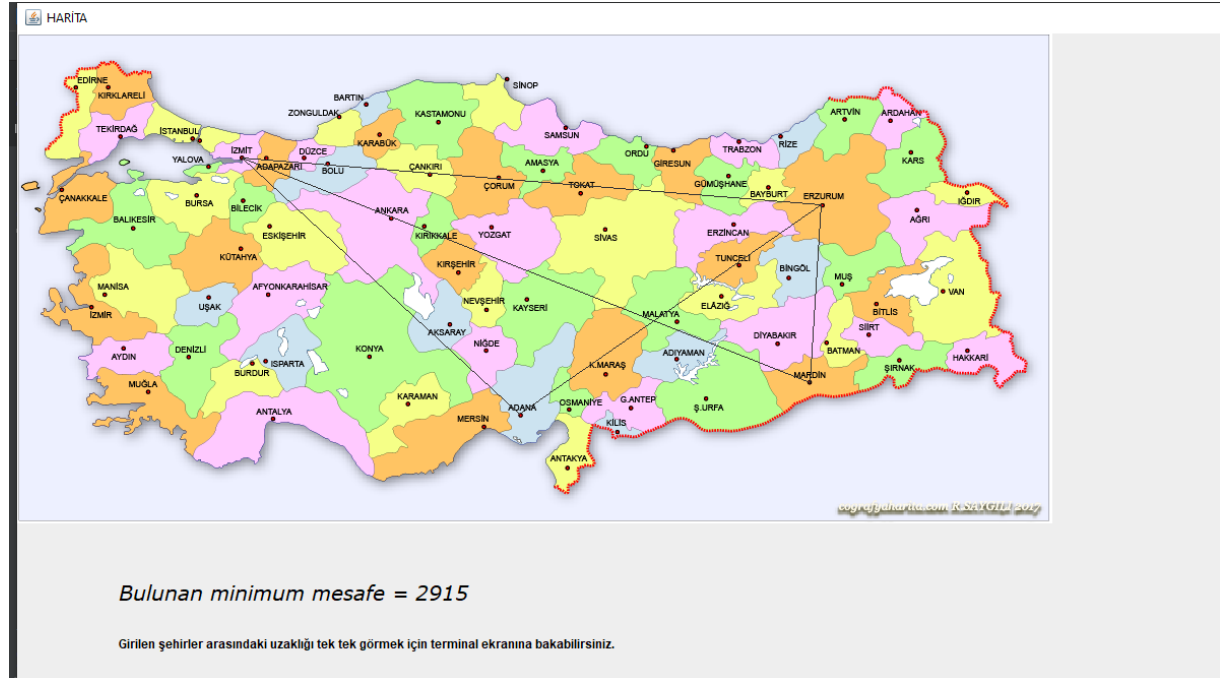
Şehri ekle

Eklenen şehirler : [1, 25, 47]

Rota yap

1 Adana	15 Burdur	29 Gümüşhane	43 Kütahya	57 Sinop	71 Kırıkkale
2 Adıyaman	16 Bursa	30 Hakkari	44 Malatya	58 Sivas	72 Batman
3 Afyon	17 Çanakkale	31 Hatay	45 Manisa	59 Tekirdağ	73 Şırnak
4 Ağrı	18 Çankırı	32 Isparta	46 K.Maraş	60 Tokat	74 Bartın
5 Amasya	19 Çorum	33 İçel	47 Mardin	61 Trabzon	75 Ardahan
6 Ankara	20 Denizli	34 İstanbul	48 Muğla	62 Tunceli	76 Iğdır
7 Antalya	21 Diyarbakır	35 İzmir	49 Muş	63 Şanlıurfa	77 Yalova
8 Artvin	22 Edirne	36 Kars	50 Nevşehir	64 Uşak	78 Karabük
9 Aydın	23 Elazığ	37 Kastamonu	51 Niğde	65 Van	79 Kilis
10 Balıkesir	24 Erzurum	38 Kayseri	52 Ordu	66 Yozgat	80 Osmaniye
11 Bilecik	25 Erzurum	39 Kırklareli	53 Rize	67 Zonguldak	81 Düzce
12 Bingöl	26 Eskişehir	40 Kırşehir	54 Sakarya	68 Aksaray	
13 Bitlis	27 Gaziantep	41 Kocaeli	55 Samsun	69 Bayburt	
14 Bolu	28 Giresun	42 Konya	56 Siirt	70 Karaman	

7.5. Haritada yolun gösterilmesi ve minimum mesafenin yazdırılması:



Kaba kod :

1. Projede verilen url'ye uygun olarak hazırlanmış dosyadaki verileri oku.
2. Okunan verileri komşuluk matrisi biçiminde oluştur.
3. Grafin oluşturulması için sınıfa git komşulukları bu matrise ekle.
4. Kullanıcıdan gitmek istediği şehirleri al.
5. Şehirleri dijkstra algoritmasının içine gönder.
6. Minimum mesafeyi bul.
7. Kullanıcının girdiği şehirler arasındaki minimum mesafelerin toplamını bul
8. Rota yap butonuna tıkladıktan sonra kullanıcının girdiği şehirlerle uyumlu olacak şekilde haritada yolu göster.
9. Haritanın geldiği ekranda aynı zamanda bulunan minimum mesafeyi yazdır.

Kaynakça

(tarih yok). Baeldung: <https://www.baeldung.com/java-graphs> adresinden alındı

Çobanoğlu, B. (tarih yok). *Yeni Başlayanlar İçin Java*. Abaküs Kitap.

GeeksforGeeks. (tarih yok). <https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-in-java-using-priorityqueue/> adresinden alındı

Şeker, Ş. E. (tarih yok). *Bilgisayar Kavramları*.

<http://bilgisayarkavramlari.sadievrenseker.com/2011/06/01/swing-kutuphanesi/> adresinden alındı

Yazılım Mutfağı. (tarih yok). <http://www.yazilimutfagi.com/index.php/2010/07/12/java-grafik-kutuphanesi/> adresinden alındı