GEZGİN KARGO PROBLEMİ

Şevki Karagöl

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

sevkikaragol@outlook.com

ÖZET- Bu projede, projeyi yapan kişi için veri yapıları ve veri modellerini anlamak ,graf yapısının kullanılmasını ve algoritma mantığını kullanarak bir probleme çözüm sağlayabilmesi amaçlanmaktadır. Bu amaç doğrultusunda öğrenciden, merkezi Kocaeli'de bulunan yeni bir kargo firmasının, siparişlerini en kısa yoldan ulaştırmasını ve tekrar Kocaeli'ye dönmesini sağlayan rotaların bulunmasını ve bu rotaların harita üzerinde gösterilmesini sağlayan bir program tasarlaması beklenmektedir.

Bu amaçlar ve isterler doğrultusunda bir rota bulma uygulaması tasarlanmıştır.Proje için gerekli veriler toplandıktan sonra veri modeline yerleştirilmiş ve proje isterleri yerine getirilmiştir.

Anahtar kelimeler-

proje,ister,rota,veri ve program

I.GİRİŞ

Proje klasörün içinde bulunan "komsuluk.txt" dosyasından şehirler ve komşularına olan uzaklıkları okunarak veri modeline yerleştirilir. Ardından "sehirPikselleri.txt" okunur ve projenin görsel kısmında kullanılacak veriler,veri modeline yerleştirilir ve kullancıdan veri girişi istenerek,kullanıcının projede etkin rol oynaması sağlanır.Proje kapsamında istenen işlemlerin gerçekleştirilmesinin ardından harita üzerinde kullanıcıya rotalar gösterilir ve proje sonlandırılır.

Bu projede veri yapıları,dosya işlemleri ve Java Swing'in bir arada kullanımına yönelik bir çalışma gerçekleştirilmiştir.Aynı zamanda öğrencilerin, proje isterlerlerinin çözümününe yönelik oluşturduğu algoritmaları ide aracılığıyla bilgisayar ortamına aktarması amaçlanmıştır.Proje için tercih edilen programlama dili "Java"olmuştur.

II.TEMEL BİLGİLER

Bu proje Java programlama dilinde geliştirilmiş olup, geliştirme ortamı olarak "NetBeans IDE ver. 8.2" kullanılmıştır.İlk etapta proje tanıtım toplantısına katılım sağlanarak gerekli notlar alınmıştır ve bir yol haritası çıkarılarak ön hazırlık sürecine girilmiştir. Bu aşamada projenin isterlerine ,kullanılacak algoritmaya,projenin yazım aşamasında hangi dilin kullanılacağına yönelik araştırmalar yapılmış olup elde edilen veriler doğrultusunda projenin ana hatları ortaya çıkarılmıştır.Projeye ilk olarak "C" dilinde başlanmıştır fakat projenin görsel kısmı hesaba katıldıktan sonra "Java" diliyle yazılmasına karar verilmiştir.Bu karar projedeki ilk zor adım olarak göze çarpmaktadır.Problemler çözüme kavuşturulduktan sonra derleyici ortamında

projenin ilk adımları atılmıştır.Proje yaklaşık 15 gün gibi süre içerisinde tamamlanmıştır.

III.YÖNTEM

Projede Dijkstra algoritması kullanılmıştır.Öncelikle Dijkstra algoritmasına Kocaeli gönderilmiştir ve diğer şehirlerin Kocaeli'ye olan uzaklıkları bulunmuştur.Buna ek olarak algoritma temelinde bulunan düğüm gezme ve işaretleme kısmına, uğradığımız düğümleri başka bir arrayliste ekleyebilmek adına gerekli kod parçası eklenmiştir.Bunu yapmamızdaki amaç ana düğümden diğer düğümlere giderken uğradığımız ara düğümleri görebilmektir.Ara düğüm bulma işlemi projede kullandığımız yöntemin temelini oluşturmaktadır.Ara düğümleri başka bir arrayliste ekleme işlemi projenin yazım aşamasında zorlanılan bir diğer işlem olmuştur.Arraylistlerde bulunan silme işlemi sayesinde bu problem çözülmüştür.

Projede kullanılan yöntem şu şekildedir.Öncelikte kullanıdan veri girişi alınır ve teslimat yapılacak şehirler bir arrayliste toplanır.Şehirler arrayliste ekleme sırasına bakılarak ilk durak olarak değerlendirlir.Örneğin Ankara,Eskişehir ve Samsun girilmiş olsun.Ankara ilk girilen şehir olduğu için Kocaeli'den sonra ilk durağımız Ankara olacaktır.Bir diğer rotada ikinci sırada girilen Eskişehir ilk durağımız olacaktır. Bu işlemi yapmamızdaki amaç rota çeşitliliği sağlamak ve alternatif rotalar bularak aracımıza en kısa yoldan teslimatları tamamlatmaktır. Örnek olarak verdiğimiz veri girişi üzerinden anlatıma devam edelim.Kocaeli'den sonra Ankara'ya gittiğimizi varsayalım.Kocaeli'den Ankara'ya giderken uğradığımız şehirler arasında teslimat yapacağımız diğer iller bulunuyorsa, bu iller teslimat yapılmış olarak(true) işaretlenir ve program çalışmaya devam eder.Ankara'ya geldiğimizde Kocaeli-Ankara arasındaki şehirler rota arraylistimize,iki şehir arasındaki uzaklık ise mesafe matrisimize eklenir.Ardından Ankara,Dijkstra algoritmasına gönderilir ve diğer şehirlere olan uzaklıkları bulunur.Teslimat yapılması gereken iller arasından Ankara'ya en yakın il bulunur ve bir önceki adımda gerçekleştirilen işlemlerin aynısı uygulanır.Böylelikle rotalar ve rotaların uzunlukları bulunmuş olur.Rotaların bulunma işlemi tamamlandıktan sonra rotalar,en kısa mesafeden en uzun mesafeye doğru sıralanır ve kullanıcıya görsel olarak sunulur.Bu sunumu gerçekleştirmek amacıyla drawLine fonksiyonu kullanılmıştır. Görsel tasarım aşamasında NetBeans eklentilerinin kullanılmasına ek olarak override işlemleri yapılarak gerekli fonksiyonlar projeye dahil edilmiştir.Projenin tasarım aşaması da sonlandırıldıktan sonra kontrol çalışmaları yapılmıştır ve projenin başka bilgisayarda açıldığında, haritanın bir kısmının görülmediği tespit edilmiştir.Bu problem projenin tamamlanma süresini biraz uzatmıştır.Projede kullanılan haritanın çözünürlüğü düşürülerek ve şehir koordinatları tekrar bulunarak bu problem çözülmüştür.Karşılaşılan problemlerin çözülmesinin ardından kontrol aşaması da son bulmuştur ve proje teslim edilmeye hazır hale gelmiştir.

Projede kullanılan fonksiyonlar:

1)"int minMesafe(int uzaklik[], Boolean dKontrol[])"

Dijkstra algortimasında kullanılan bir fonksiyondur.Henüz uğranmamış ve en yakın mesafede bulunan düğümü bulmak için çalıştırılır.

2)" void dijkstra(int graf[][], int dugum)"

Projemizin temelini oluşturan fonksiyondur.Bu fonksiyon yöntem kısmında bahsettiğimiz işlemleri gerçekleştirir ve rota bulmada kullanılmıştır.İki şehir arasındaki durak şehirleri ters bulduğu için "Collections.reverse" kullanılarak rota ters çevrilmiştir.

3)" int minBul(Boolean kontrol[])"

Bir şehre teslimat yaptıktan sonra teslimat yapılacak diğer şehirler arasından en yakın şehri bulmak için kullanılır.

4)" ArrayList clearFonks(ArrayList a)"

Dijkstra fonksiyonuna şehir gönderme işleminden önce kullanılır.Bu fonksiyonu kullanma sebebimiz bir önceki şehir gönderme işlemi sonucu oluşan arraylistleri temizlemektir. Eğer temizleme işlemi yapılmazsa yeni veriler eski verilerin üzerine yazılmaya çalışılmaktadır.Bu durum programın hata vermesi neden olur.

5)" ArrayList rotaEkle(int plaka,int sayac)"

Teslimat yapılacak şehirleri ve bu şehirlere giderken uğranan ara şehirleri,kullancıya sunulan rotalara ekleme işlemini gerçekleştirir.

6)" arrlKiyas(ArrayList a,ArrayList b)"

Kullanıcıya sunulacak alternatif rotaların hepsinin birbirinden farklı olmasını sağlar.İki rota arraylistinin karşılaştırılması ve aynı ise birinin silinmesi prensibiyle çalışır.

7)" void main(String[] args)"

Projemizin ana fonksiyonudur.Dosya okuma, kullanıcıdan veri girişi yapılması,bulunan rotaların görüntülenmesi ve rota görüntüleme ekranın açılması işlemi burada yapılır.

8)" void paint(Graphics g)"

Rota görüntüleme ekranında bulunan çizgilerin çizilmesini sağlayan fonksiyondur.Şehirlerin koordinatlarından faydalanarak iki şehir arası çizgi çizilmesini sağlar.

Projede kullanılan görsel tasarım elemanları:

1)JPanel

2)Frame

3)Label

4)OptionPane

IV.SÖZDE(PSEUDO) KOD

Verilen proje için hazırladığımız programın pseudo kodları şu şekildedir:

1-Program çalıştı.

2-Program, içerisinde yer aldığı dosya yolunda

bulunan "komşuluk.txt" ve "sehirPikselleri.txt" dosyalarını okudu ve veri modellerine işledi.

3-Kullanıcıya,kaç şehre teslimat yapılacağı soruldu.

4-Kullanıcı kaç şehre teslimat yapılacağını girdi. **ek1*

5-Kullanıcıya,hangi şehirlere teslimat yapılacağı soruldu.

6-Kullanıcı teslimat yapılacak illeri programa girdi. **ek2*

7-Kullanıcıya teslimat yapmak istediği illeri içeren rotalar sunuldu. **ek3*

8-Kullanıcıya rotaların görselleştirilmiş halleri sunuldu. **ek4*

9-Kullanıcı ekranda bulunan butonun üzerine tıkladı. **ek5*

10-Alternatif rota kullanıcıya gösterildi. **ek6*

11-Kullanıcı kendisine sunulan rotalar bitene kadar butona tıklama işlemini gerçekleştirdi ve tüm rotaları gördü.

12-Rotaların bittiğini kullanıcıya bildiren mesaj ekrana geldi. **ek 7*

13-Kullanıcı "ok" butonuna basarak programı sonlandırdı.

V. EKLER

***ek 1**

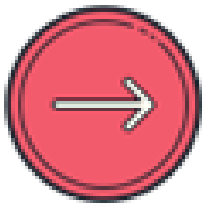
```
!!!Sehir isimlerini girerken ilk harfi buyuk yaziniz ve Turkce karakter kullanmayiniz!!!
Sehir isimlerini tam ve aralarinda birer bosluk birakarak giriniz ya da alt alta giriniz.


        Kac sehire teslimat yapacaksiniz? :10
```

***ek 2**

```
Hangi sehirlere teslimat yapacaksiniz? :
Ankara
Bursa
Rize
Mus
Van
Agri
Samsun
Balikesir
Hatay
Hakkari
```

***ek 5**

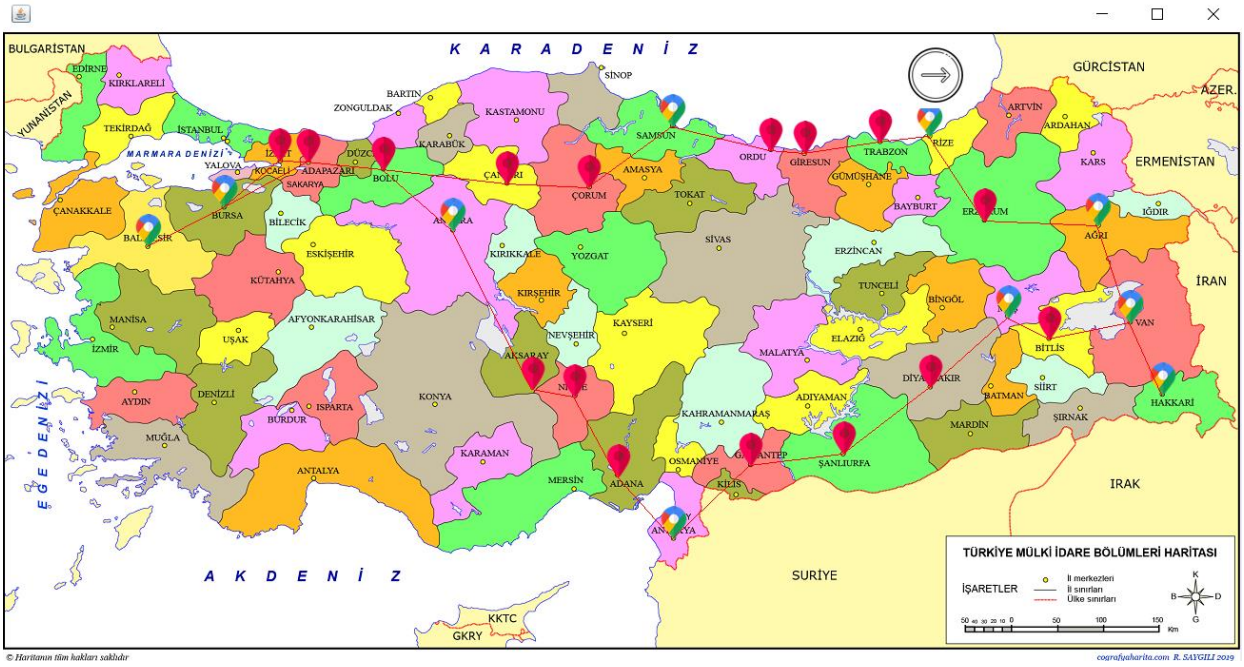


***ek 3**

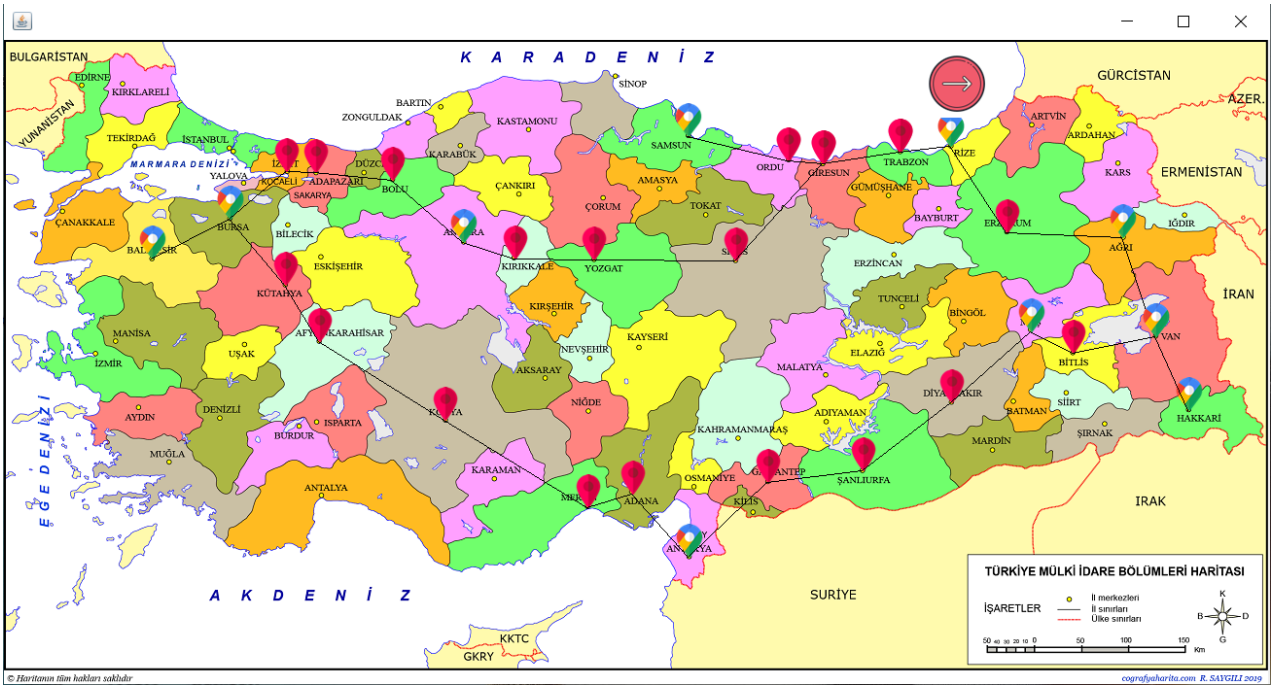
```
Teslim yapacaginiz sehirler için rotalar ↓↓↓
→Kocaeli→Sakarya→Bolu→Ankara→Bolu→Sakarya→Kocaeli !!Mesafe-> 684 kmdir..
```

Ekrana sığması için kısa bir rota seçilmiştir.

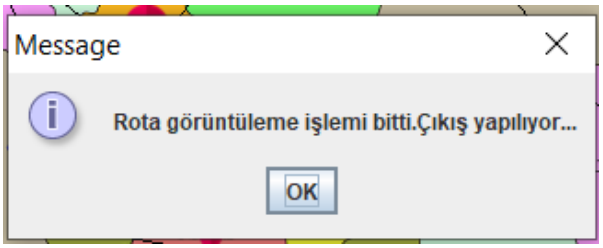
***ek 4**



***ek 6**



***ek 7**



VI.KAYNAKÇA

[1]howtodoinjava.com

[2]youtube.com

[3]udemy.com

[4]edestek.kocaeli.edu.tr

[5]javahungry.blogspot.com

[6]stackoverflow.com

[7]geeksforgeeks.com

[8]flaticon.com