

GEZGİN KARGO PROBLEMİ

Sefa Mert GÜNGÖR
Bilgisayar Mühendisliği
180201086
KOCAELİ ÜNİVERSİTESİ
meertgngnr5255@hotmail.com

Miraç Onur SAKA
Bilgisayar Mühendisliği
180201081
KOCAELİ ÜNİVERSİTESİ
m.onur7@hotmail.com

Özetçe- Bu projemizde dijkstra yol algoritmasını kullanarak, Kocaeli’den başlayarak, istenilen maksimum 10 şehre uğramak üzere en kısa 5 yolu bulduk ve gösterdik.

Anahtar kelimeler- dijkstra, swing, komşuluk, şehir, uzaklık

I. GİRİŞ

Proje klasörümüzün içinde bulunan “sehirler.txt” ve “u2.txt” dosyalarından şehirlerin isimlerini ve birbirine olan uzaklıklarını okuyarak veri modeline yerleştirdik ve proje içinde istenen işlemlerimizi gerçekleştirdik. Yapılan bu işlemlerden sonra veri modelimizdeki bilgilerle dijkstra algoritmasını kullanarak bizden istenen kargo programımızın başlangıcını yaptık Proje için kullanılmaya uygun görülen dil Java’yı kullandık ve Java’nın swing kütüphanesinden de yararlandık.

II. HAZIRLIKLAR VE BİLGİLER

Bu proje Java programlama dilinde geliştirilmiş olup, geliştirme ortamı olarak “Apache Netbeans 11.1” kullanılmıştır. İlk önce bizden istenen durumları değerlendirip tartışarak ne yapmamız gerektiği hakkında karara varıp projeyi hangi adımlara göre takip edeceğimize karar verdik. Daha sonrasında projeye başladık.

III. YÖNTEM

İlk olarak “u2.txt” den okuduğumuz şehirler arasındaki mesafeleri uzaklık matrisimize attık. İkinci olarak “sehirler.txt” deki şehirlerimizin isimlerini sıralı olarak şehirlerİsim dizimize attık. Daha sonra EnKısaYol objesi oluşturarak class’ın içine yazdığımız dijkstra algoritması sayesinde 2 şehir arasındaki en kısa mesafeyi bulabilmiş olduk. Daha sonra yolları bulabilmemiz için gidilecekŞehirler adlı fonksiyonumuzu yazdık bu fonksiyonumuz sayesinde de 5 tane en kısa yol bulabildik. Daha sonra EnKısaYol classının içinde bulduğumuz yolları, oluşturduğumuz 5 farklı JFrame Formuna atarak görselliği tamamlamış olduk.

Projede kullanılan fonksiyonlar:

“public static int plakaBulma()” girilen şehir isimlerini plaka olarak döndüren fonksiyondur.

“int mesafe()” Bir şehrin komşularına olan uzaklıkları arasındaki en kısa mesafeli şehrin indexini bulan fonksiyondur.

“void gidilecekSehirler()” gerekli bilgileri alarak en kısa 5 yolu buldurmayı sağlayan fonksiyondur.

“void dijkstra()” gidilecek şehirler fonksiyonundan aldığımız bilgilerle en kısa istikameti bulmamızı sağlayan fonksiyondur.

“public void paint()” Açılan JFrame penceresinde resmi, şehirler arası çizgileri ve kaçınıcı sırada olduklarını gösteren fonksiyondur.

“private void jButton1ActionPerformed()” Bütün haritalar classlarında bulunur. Bir sonraki haritanın açılmasını sağlayan fonksiyondur.

IV. YALANCI KOD

- 1- Program çalıştırılır.
- 2- u2.txt” den şehirlerin birbirine olan uzaklıkları okunur.
- 3- “sehirler.txt” den şehirlerin ismi okunur
- 4- Kaç şehre gidileceği kullanıcı tarafından alınır.
- 5- Hangi şehirlere gidilecekse isimleri kullanıcı tarafından alınır.
- 6- EnKısaYol sınıfından cisim oluşturulup bazı bilgiler gönderilir.

7- EnKısaYol sınıfında dijkstra algoritması en kısa yolu çıkartır.

8- Döngüdeki dijkstra algoritmasından en kısa 5 yol çıkartılır.

9- Gidilen yollar diziye atılır.

10- Atılan yollar output penceresine sıra sıra çıkartılır ve görüntüye işlenmek için JFrame formu olan Harita classına cisim oluşturularak gönderilir.

11- Harita classında gidilen yollar çizgi çekilerek gösterilir.

12-Çizilen yollar gezme sırası ile numaralandırılır.

13-Harita üzerindeki butonlara tıklayarak numaralandırılan diğer harita sınıflarındaki cisimlerin çağırılması sağlanır.

V.KULLANILAN KÜTÜPHANELER

```
import java.awt.RenderingHints;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.DataInputStream;
import java.io.FileInputStream;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.Scanner;
import java.util.ArrayList;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.geom.Line2D;
import java.awt.Color;
import java.awt.BasicStroke;
```

Ekler

Kaç şehir girileceği buraya yazılıyor:

Kac şehir girilecek:
5

Hangi şehirlere gidileceği buraya yazılıyor:

Sehirlerin plakaları neler:
Ankara
Antalya
Izmir
Samsun
Zonguldak

Gidilen şehirler tek tek sağdan başlayarak yazılır.

Bulunan 1. yol:

Ankara Bolu Sakarya Kocaeli

Antalya Konya Ankara

Izmir Aydın Denizli Burdur Antalya

Samsun Corum Kirikkale Ankara Eskişehir Afyonkarahisar Uşak Manisa İzmir

Zonguldak Karabük Çankırı Corum Samsun

Kocaeli Sakarya Düzce Zonguldak

Toplam alınan yol:3283

40 53 13 5 41 6 14 19 8 34 44 63 2 25 5 70 18 54 18 17 77 66 80 53 40

Bulunan 2. yol:

Antalya Isparta Afyonkarahisar Eskisehir Bilecik Sakarya Kocaeli

Izmir Aydın Denizli Burdur Antalya

Samsun Corum Kirikkale Ankara Eskisehir Afyonkarahisar Usak Manisa Izmir

Zonguldak Karabük Cankiri Corum Samsun

Kocaeli Sakarya Düzce Zonguldak

Toplam alınan yol:3000

40 53 10 25 2 31 6 14 19 8 34 44 63 2 25 5 70 18 54 18 17 77 66 80 53 40

Bulunan 3. yol:

Izmir Balıkesir Bursa Kocaeli

Samsun Corum Kirikkale Ankara Eskisehir Afyonkarahisar Usak Manisa Izmir

Zonguldak Karabük Cankiri Corum Samsun

Antalya Konya Ankara Bolu Zonguldak

Kocaeli Bursa Kutahya Afyonkarahisar Burdur Antalya

Toplam alınan yol:3509

40 15 9 34 44 63 2 25 5 70 18 54 18 17 77 66 13 5 41 6 14 2 42 15 40

Bulunan 4. yol

Samsun Corum Cankiri Bolu Sakarya Kocaeli

Zonguldak Karabük Cankiri Corum Samsun

Ankara Bolu Zonguldak

Izmir Balikesir Kutahya Eskisehir Ankara

Antalya Mugla Aydın İzmir

Kocaeli Bursa Kutahya Afyonkarahisar Burdur Antalya

Toplam alınan yol:3428

40 53 13 17 18 54 18 17 77 66 13 5 25 42 9 34 8 47 6 14 2 42 15 40

Bulunan 5. yol

Zonguldak Düzce Sakarya Kocaeli

Ankara Bolu Zonguldak

Izmir Balikesir Kutahya Eskisehir Ankara

Antalya Mugla Aydın İzmir

Samsun Corum Kirikkale Kirsehir Aksaray Konya Antalya

Kocaeli Sakarya Bolu Cankiri Corum Samsun

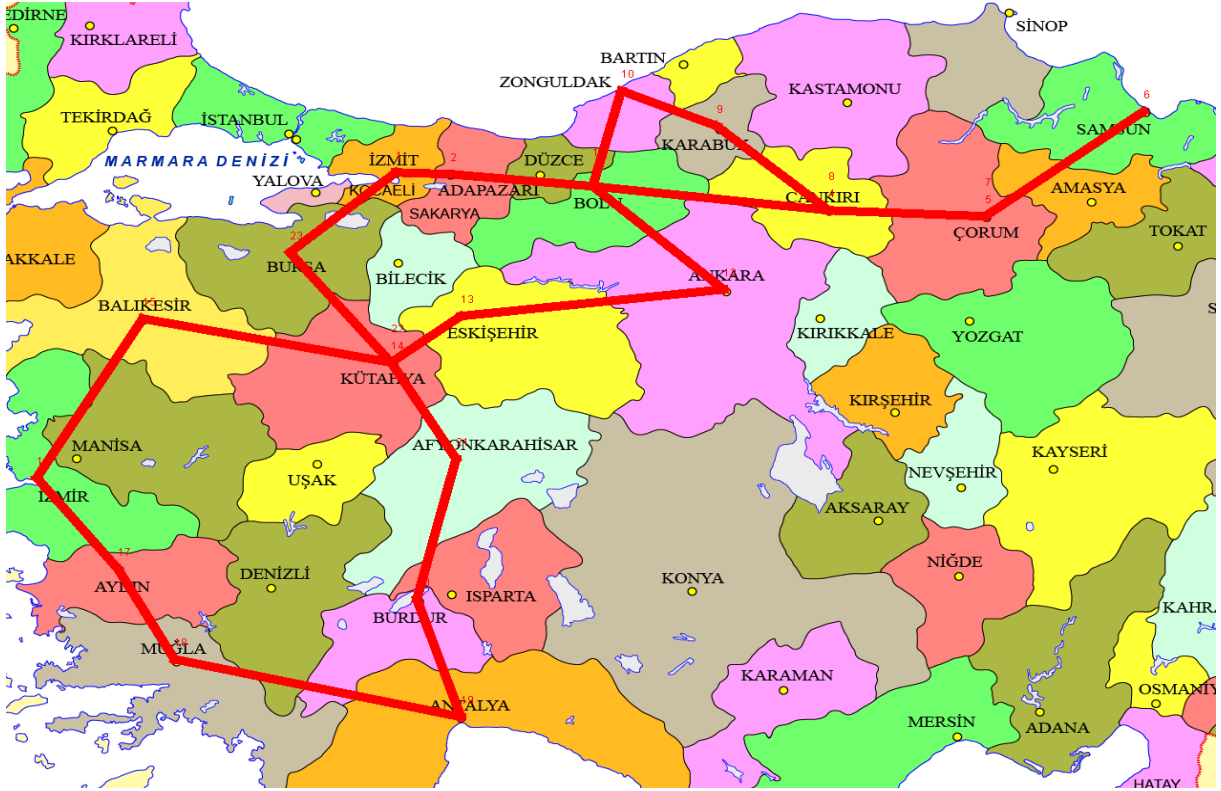
Toplam alınan yol:3428

40 53 80 66 13 5 25 42 9 34 8 47 6 41 67 39 70 18 54 18 17 13 53 40

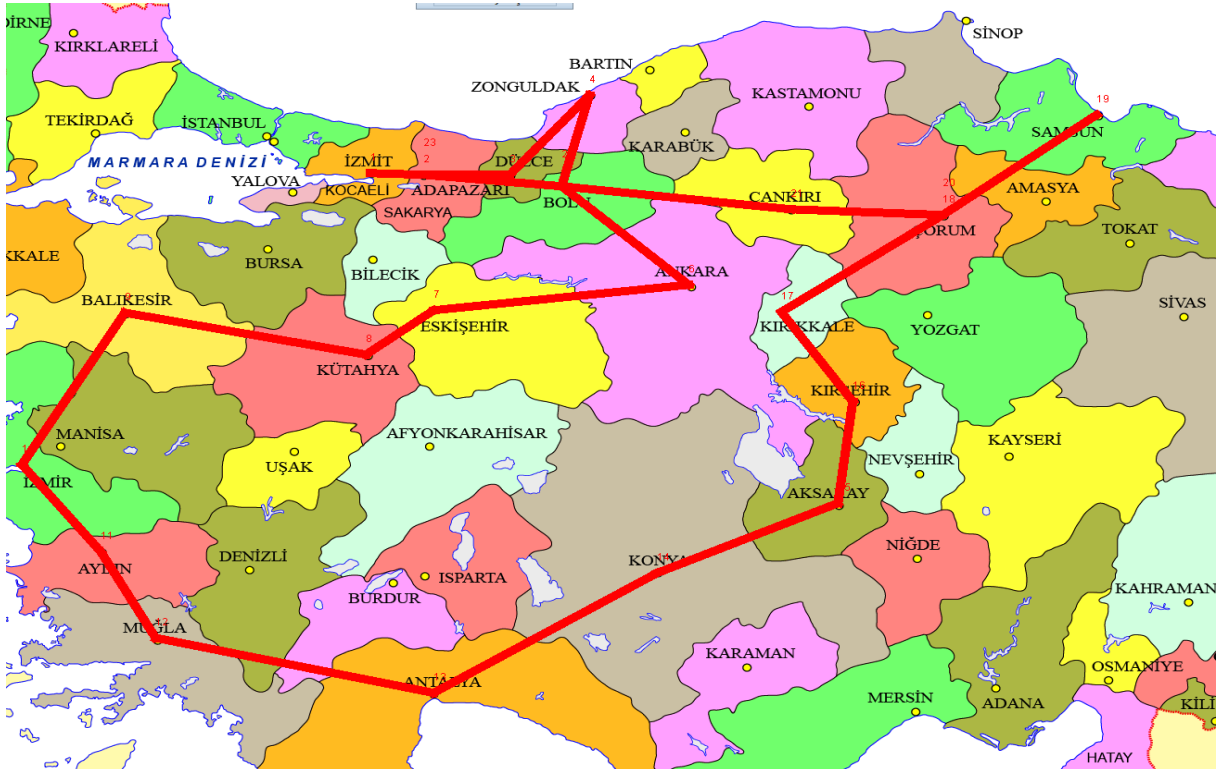
BUILD SUCCESSFUL (total time: 5 minutes 29 seconds)

Görselde bulunan yollar en kısa mesafeli yoldan uzuna doğru sıralanarak gösterilir.

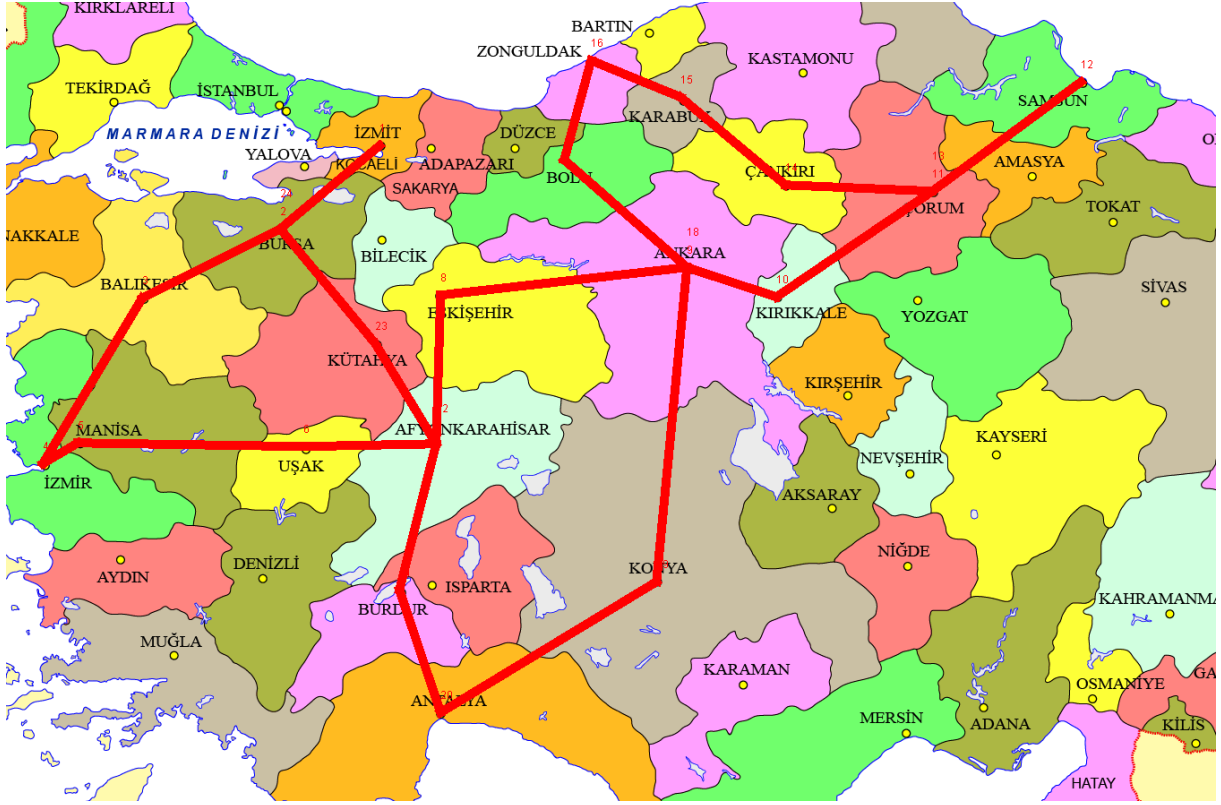
En kısa yol:



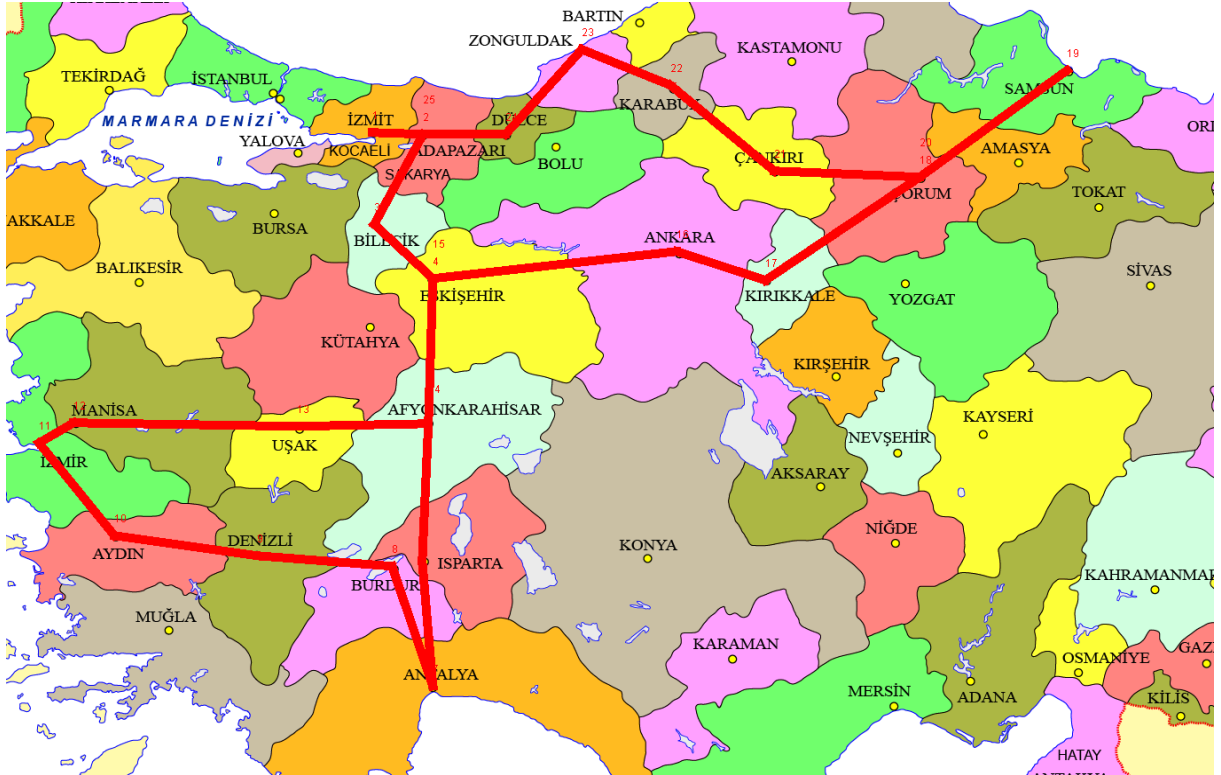
En kısa 2. yol:



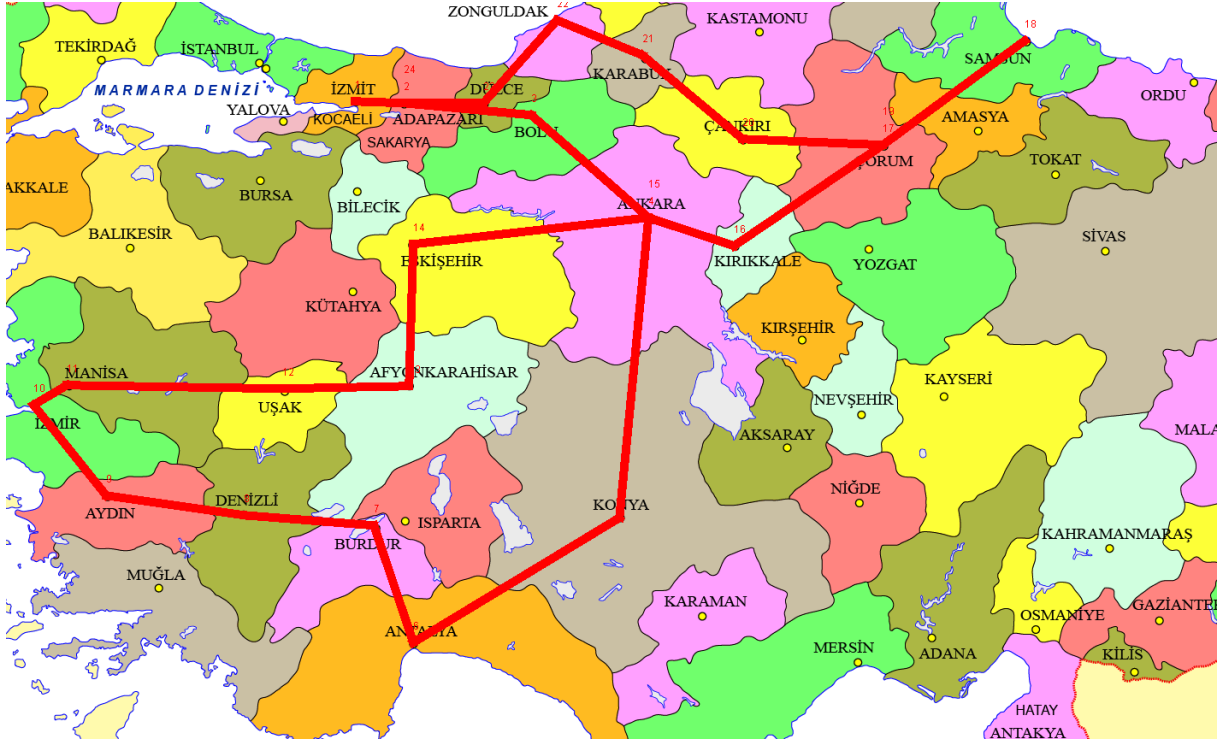
En kısa 3. yol:



En kısa 4. yol:



En kısa 5. yol:



KAYNAKÇA

<https://www.youtube.com/watch?v=q8n2oaokrcU>

https://www.oocities.org/turhan_coban/Java/kitap/java06.htm?202024#6.5

<http://bilgisayarkavramlari.sadievrenseker.com>

<https://www.geeksforgeeks.org>