**TRAFİK AKIŞ PROBLEMİ**

Yaşar Can ÇİLİNGİR

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

[yasarcanccd@gmail.com](mailto:yasarcanccd@gmail.com)

**ÖZET**

Programım, kullanıcıya sunulan iki haritadan birisini seçmesi ile başlar. Daha sonra x,y,z,t yollarından giriş ve çıkış olanların kullanıcı tarafından belirlenmesi ile devam eder. 2’den fazla giriş ve çıkış belirlenemez. Kullanıcıdan yoğunluk bilgileri ve giriş veya çıkış olmayan yolların yön bilgisi alınır. (Programın sorunsuz çalışması için yönler ile giriş çıkışlar tutarlı olmalıdır). Yoğunluk bilgisi bilinmeyen yol değeri için -1 girilmelidir. Kullanıcıdan alınan yoğunluğu bilinmeyen yol sayısına göre matris oluşturularak bilinmeyenlerin bulunduğu denklemler aktarılır. Bilinmeyen denklemlerin aktarılmış olduğu matrise elementer işlemler uygulanarak Gauss Jordan çözümü uygulanır.

**1.GİRİŞ**

Gauss Jordan, lineer denklemlerden oluşan problemin çözümünü bulmak amacıyla kullanılan yöntemdir. Çok bilinmeyenli denklemleri çözerken büyük fayda sağlamaktadır. Gauss Jordan yöntemi, bilinmeyen katsayıları matris şeklinde gösterilen denklem sistemini çözer. Çözümün kesin olması için bilinmeyen sayısı denklem sayısından fazla olmamalıdır.

Program, yoğunluğu bilinmeyen yolları (yoğunluğu -1 girilen yollar) diğer yolların yönleri ve yoğunlukları ile bağlantı kurup denklem haline getirir. Bu denklemleri matriste göstererek Gauss Jordan ile çözer.

**2.TEMEL BİLGİLER**

Program C++ programlama dilinde geliştirilmiş olup geliştirme ortamı olarak Microsoft Visual Studio 2015 kullanılmıştır.

**3. YÖNTEM**

Programlanma aşamaları altta verilen başlıklar altında açıklanmıştır.

**3.1 ALGORİTMA**

Program öncelikle kullanıcıya sunulan iki haritadan birisini seçmesini isteyecek. Seçilen haritada kullanıcı x,y,z,t yollarından giriş ve çıkışları belirleyecek. Giriş ve çıkış olarak belirlenen yollar, daha sonra denklemleri kurarken kontrol edilmek amacıyla girisler[] ve cikislar[] isimli dizilerde kayıt edilir.

Kullanıcıdan yol yoğunlukları alınır, bilinmeyen yola ‘-1’ değerini girmesi istenir.

Kullanıcıdan giriş çıkış olmayan yolların yönleri alınır. Çözümün doğru olabilmesi için yönler ile giriş çıkış olan yollar tutarlı olmalıdır. Alınan yönler ve değerlere göre denklemler kurularak matrise aktarılır.

Kurulan matris Gauss Jordan yöntemine göre elementer işlemler ile birim matris hale getirilerek çözülür.

**3.2 Denklemler**

int yollar[] dizisi oluşturulmuştur.

denklemlerde

**yollar[0] => x, yollar[1] => y, yollar[2] => z, yollar[3] => t, yollar[4] => a, yollar[5] => b, yollar[6] => c, yollar[7] => d, yollar[8] => e**

**yollarını temsil etmektedir**

**Örnek denklem:**

**1. Harita, x çıkış,**

**a yolunun başlangıç noktası t, yön noktası x**

**b yolunun başlangıç noktası y, yön noktası x**

durumu için

cozummatrisi[1] = yollar[4] + yollar[5] - yollar[0];

denklemi kurulur. Yollardan birinin bilinmeyen(-1) girilmesi durumunda sonucu etkilememesi amacıyla cozummatrisi[] değiştirilir. Örnek olarak yukardaki örnekte a yolu bilinmeyen(-1) ise

cozummatrisi[1] değeri 1 artırılır.

Bilinmeyen olarak girilen yolları 2 boyutlu yollarklon[][] dizisine aktarılarak bilinmeyen matrisi oluştulur. Yukardaki örnek için,

**yollarklon[1][4] = yollar[4];**

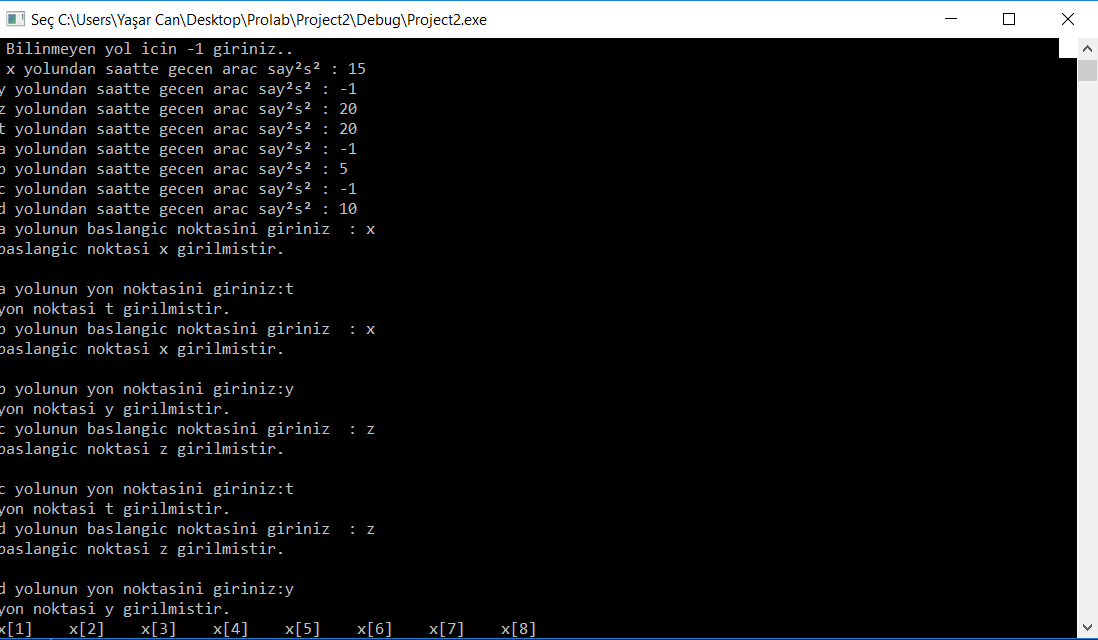
işlemi yapılır.

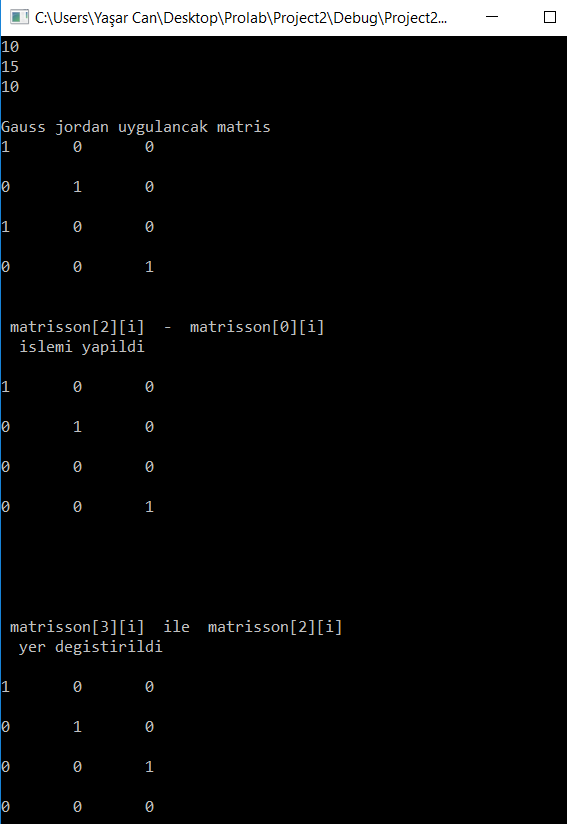
**4. YALANCI KOD**

* İki haritadan birisini al.
* X, Y, Z, T yollarından giriş ve çıkış olanları al.
* Giriş ve çıkış dışındaki yolların yoğunluklarını al.
* Bilinmeyen yollar için -1 değerini al.
* Giriş ve çıkış dışındaki yolların referans noktalarına göre yönlerini al.
* Yönü kontrol et.
* Girişleri ve çıkışları kontrol et.
* Kontrol edilen yön, giriş ve çıkış şartlarını sağlayan denklemi çöz.
* Denklemdeki temsil edilen bilinmeyen yolu iki boyutlu diziye aktar.
* Denklem çözümünü tek boyutlu diziye aktar.
* İki boyutlu dizide bilinmeyenin olduğu sütunları belirle.
* Belirlenen sütunları, boyutları bilinmeyen sayısı olan iki boyutlu diziye aktar.
* Gauss Jordan çözülecek matris oluşturulur.
* Matrisin sol alt üçgeni sıfırlanır
* 0. Sütunu, 1. Satırdan itibaren 0. Satır kullanarak sıfırla
* 1. Sütunu, 2. Satırdan itibaren 1. Satır kullanarak sıfırla
* i. Sütun i. Satırdan itibaren i. satır kullanarak sıfırla
* Matrisin sağ üst üçgeni sıfırlanır
* 0. Satırı, 1. Sütundan itibaren 1. Satır kullanarak sıfırla
* 1. Satırı, 2. Sütundan itibaren 2. Satır kullanarak sıfırla
* Çözüm matrisine satır işlemlerinin aynısını yap.
* Birim matris formuna ulaşılır.

**5. SONUÇLAR**

Yönler, giriş ve çıkışlar ile tutarlı olarak girildiğinde, kesin sonucu olan bilinmeyenler girildiğinde program doğru çalışmaktadır. Çözülemeyen denklem kontrolü proje isterlerinde olmadığı için yapılmamaktadır, sonuç matrisi birim matris değilse denklemler çözülemez demektir.



****