**KOCAELİ ÜNİVERSİTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**YAZILIM LABORATUVARI I DERSİ 2. PROJE**

**Proje: KARE OLMAYAN MATRİSİN SÖZDE TERSİNİ BULMA**

**Amaç:** C++ / C#ya da Java programlama dillerinden birini kullanarak [**M**]x[**N]** (**M**≠**N**) boyutlarında bulunan bir matrisin sözde tersinin (pseudo inverse ) alınması beklenmektedir.

Eğer matrisiniz kare değilse ters matrisi elde edemezsiniz. Bunun yerine, sanki ters matrismiş gibi bir matris veren yöntemler kullanılabilir (En küçük kareler yöntemi (Least Squares Method), Tekil Değer Ayrışımı (Singular Value Decomposition), … vb.). Kullanılacak olduğunuz algoritmanın/yöntemin hem matematiksel olarak hem de kodsal olarak nasıl hesap yaptığının tarafınızdan anlaşılmış olması beklenmektedir.

Kısaca istenen bu yöntemlerden herhangi birini kullanarak [**M**]x[**N]** boyutlarında bulunan genel bir matrisin sözde tersinin (pseudo inverse ) alınmasıdır.

* Bu amaçla [**M**]x[**N]** boyutlarında genel bir matris oluşturulacaktır. (**M** ve **N** birbirinden farklı [1-5] arasında bir rakam olmalıdır)
* Uygulama iki farklı şekilde giriş alacaktır.
  + Bunlarda ilkinde [**M**]x[**N]** boyutları rastgele olarak atanacaktır. Matris her bir elemanının değeri [1-9] arasında (tamsayı ya da virgülden sonra en fazla bir basamak olabilir) rastgele olarak atanan değerlerden oluşacaktır. Matris oluşturulduktan sonra form **(Consol ekranı değil!)** ekranında gösterilecektir.
  + İkinci giriş seçeneğinde; [**M**]x[**N]** değerleri ve matrisin elemanları elle girilerek oluşturulacaktır. Bu şekilde matris oluşturulduktan sonra form ekranında gösterilecektir.
  + Kullanıcı hangi girişi yapacağını uygulama açıldığında seçebilmelidir.
* Matrisin sözde tersi alınmalıdır (En küçük kareler yöntemi (Least Squares Method), Tekil Değer Ayrışımı (Singular Value Decomposition), … vb. yöntemler kullanılabilir)
* Formda matristeki ilk işlem adımından son işlem adımına kadar (seçilen algoritmanın/yöntemin göz önüne alınarak) gerçekleşen tüm değişimler ekranda gösterilmelidir.
* Form üzerinde matrisin sözde tersinin alınması aşamasında gerçekleştirilen tüm toplama ve çarpma işlem adet sayısı (computational complexity) gösterilmelidir:
  + Toplama sayısı = [ toplam işlemi sayısı + çıkarma işlem sayısı]
  + Çarpma sayısı = [ çarpma işlemi sayısı + bölme işlem sayısı]
* Hazır kütüphane kullanan projeler değerlendirme dışı bırakılacaktır.

**Son Teslim tarihi: 30.11.2018 Cuma günüdür. Projeyi nasıl teslim edeceğiniz ve sunum tarihleriniz daha sonra duyurulacaktır.**

**Not: Proje ile ilgili sorularınızı Arş. Gör. Burcu KIR SAVAŞ ve Arş. Gör. Hikmetcan ÖZCAN a mail yoluyla sorabilirsiniz. Projede yapmanız gerekenler açıkça belirtilmiştir. Açıklamaları dikkate alarak projenizi geliştiriniz.**