**MATRİX ALGORİTMASI**

Mert Eren Keküç

Yazılım Mühendisliği 2. sınıf

1. İlk olarak iki matrisin toplanması işlemi için olan collection isimli fonksiyonu yazdım. Bu fonksiyon 4 parametre alıyor line\_number ile matrisin satır sayısı column\_number ile de matrisin sütun sayısı alınıyor ardından iki tane de matris alıyoruz. Matrislerde toplama işlemi yapılabilmesi için iki matrisin satır ve sütun sayısı eşit olmalı bu yüzden ikinci defa satır veya sütun sayısı almıyoruz onun yerine diğer matriste de aynı satır sütun sayısını kullanıyoruz. Fonksiyonun içinde result isimli bir matris daha tanımlıyoruz toplam da bir matris olacağı için bunun içine yazdıracağız. Toplama işlemi de for döngüsü sayesine iki matrisin aynı satır ve sütunlarına denk gelen sayıların result matrisinin aynı satır ve sütununa yazılması ile yapılıyor. Ardından da result matrisi yazdırılıyor.
2. İkinci işlemimiz ise noktasal çarpım yani bir sayı ile matrisin çarpılması işlemi bu fonksiyonun ismi de multiplication. Yine collection fonksiyonunda olduğu gibi 4 parametre alıyoruz toplamada aldığımız ikinci matris yerine çarpılacak sayıyı parametre olarak alıyoruz sadece bu sefer. Yine for döngüsü yardımı ile matrisin tüm elemanlarını aldığımız sayı ile çarpıp yine result fonksiyonu içerisindeki aynı satır ve sütununa yazdırıyoruz. Son olarak yine oluşan result matrisini ekrana yazdırıyoruz.
3. Üçüncü işlemimiz determinant alma işlemi fonksiyonumuzun ismi determinant. Bu sefer iki parametre alıyoruz çünkü bir matrisin determinantının olabilmesi için satır ve sütun sayısı eşit olmalı. Bir de yine her zamanki gibi matrisimizi alıyoruz. Bu sefer farklı olarak temp isminde geçici bir matris oluşturuyoruz. Fonksiyonumuzda eğer matris 1x1 matris ise onu direkt yazdırıyoruz. Eğer değilse 1x1 den büyük olan matrisleri hesaplamak için kofaktörü kullanacağız. Aslında buradan sonraki adımlarda matrisin boyutunu küçültüp işlem yapmış oluyoruz. Burada elde ettiğimiz değerleri işlem yapabilmek için geçici matrise aktarıyoruz. Yine burada da hanoi kulelerinde olduğu gibi recursive fonksiyonları kullanıyoruz. Kofaktör yardımıyla determinant hesaplanmasında bir satır veya sütun seçilir bu satır veya sütundaki her bir eleman ile onun kofaktörünün (kendisi x -1 üzeri satır+sütün) çarpımı determinanta eşittir. Determinantın formülündeki kofaktörün hesaplanmasında kullanılan formüldeki -1 ibaresi için de s değişkeni atadık ve onu kullanıyoruz. Son olarak bu fonksiyon bir float değer döndürür.
4. Son olarak matrisin tersinin alınması ile ilgili fonksiyonu yazdım burada mecburen başka bir fonksiyon daha gerekti normalde bize kofaktör lazım yine bu kısmı determinantta da yapmıştık ama bize hem kofaktör hem determinant lazım olduğu için başka bir fonksiyon daha gerekti. Buradaki formülümüzde kofaktör matrisimizin transpozunu determinanta bölüyoruz. Bunun için önce kofaktör bulmak için cofactor isimli fonksiyonu yazdım. Determinantla çok benzer bir fonksiyon bu sefer de iki parametre alıyoruz yine determinantta olduğu gibi. Ekstra olarak yine bir geçici matris olan temp ve kofaktör halini tutabilmek için cof matrisi tanımlıyoruz. Sonuna kadar çok benzer gidiyor işlemler determinantla. Ama bunun sonunda transpose fonksiyonunu çağırıyoruz ve -1 ibaresi yerine -1 üzeri satır+sütun ibaresini math.h daki pow fonksiyonu ile ekliyoruz. Şimdi burada transpose fonksiyonundan bahsettim bu transpose isimli fonksiyonumuz 3 parametre alıyor matrisimiz , satır ya da sütun sayımız ve kofaktörü tuttuğumuz cof matrisimiz. Değişken olarak da yine bir geçici matris olan temp ve tersini yazdıracağımız matris olan inverse matrisini ekliyoruz. Burada for döngüleriyle yine geçici matrisimize kofaktör matrisimizi satır ve sütunları yer değiştirecek şekilde yazdırıyoruz. Bu kısımdan önce determinant fonksiyonumuzu burada çağırıyoruz ve 0 dan büyük olma koşulunu da ekliyoruz çünkü determinantı 0 olan matrislerin tersi olmaz. Yazdırdığımız ters matrisimizi determinanta bölüyoruz ve sonuca ulaşıyoruz. Son olarak da inverse matrise çıkan sonuçları yazdırıyoruz.

Son olarak matrix.h kütüphanesini oluşturdum ve projeye eklemek için **tam konumunu** yazdıktan sonra gerekli kodları yazıp çalıştırmamız yeterli. #include "C:\Users\merte\Desktop\Matrix.h" şeklinde ekledim.