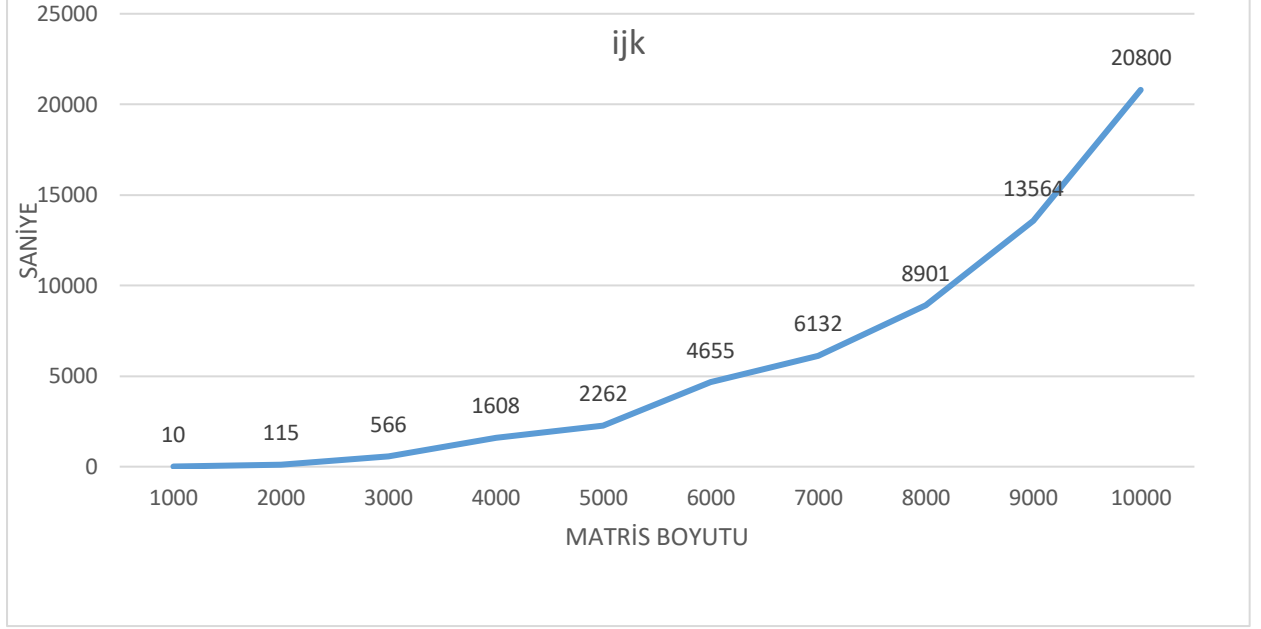




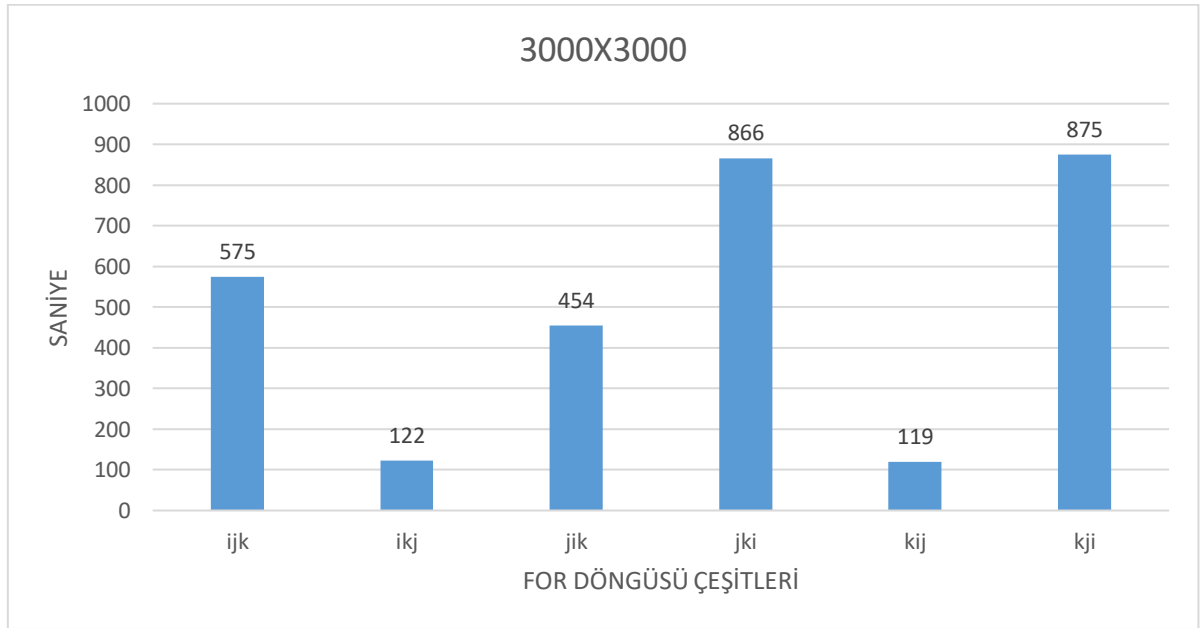
Ders adı:	Bilgisayar Programlama III
Ad Soyad:	Üveys SARAÇ
Okul numarası:	1621221025
Ödevin adı:	Matris – Matris çarpımı

GRAFİKLER

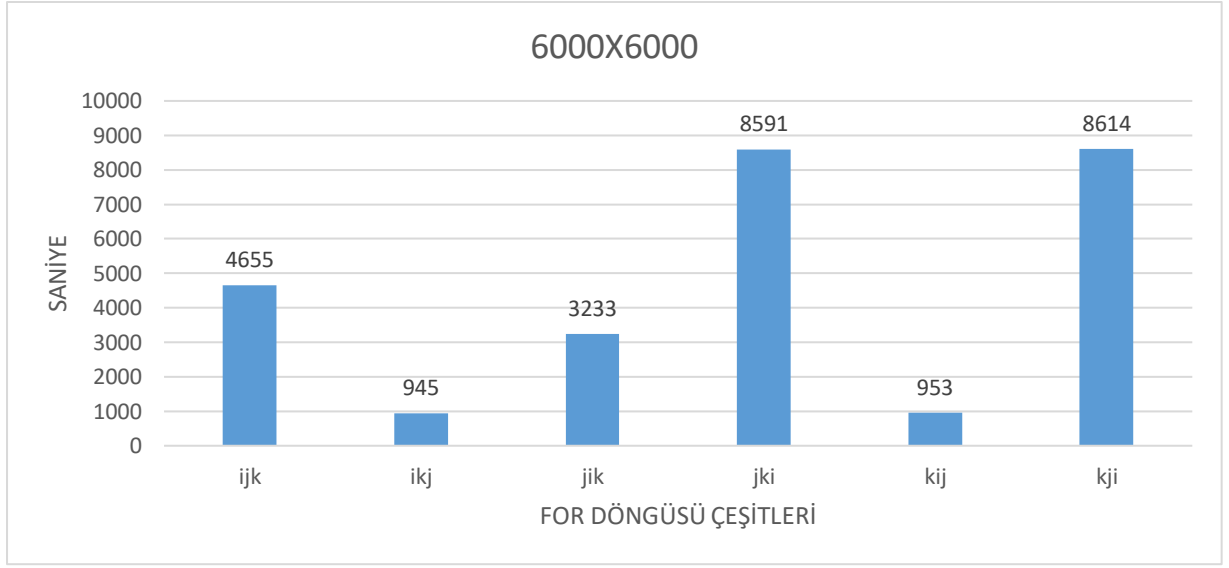
İJK TÜM MATRİS BOYUTLARI SANİYE/MATRİS BOYUTU GRAFİĞİ



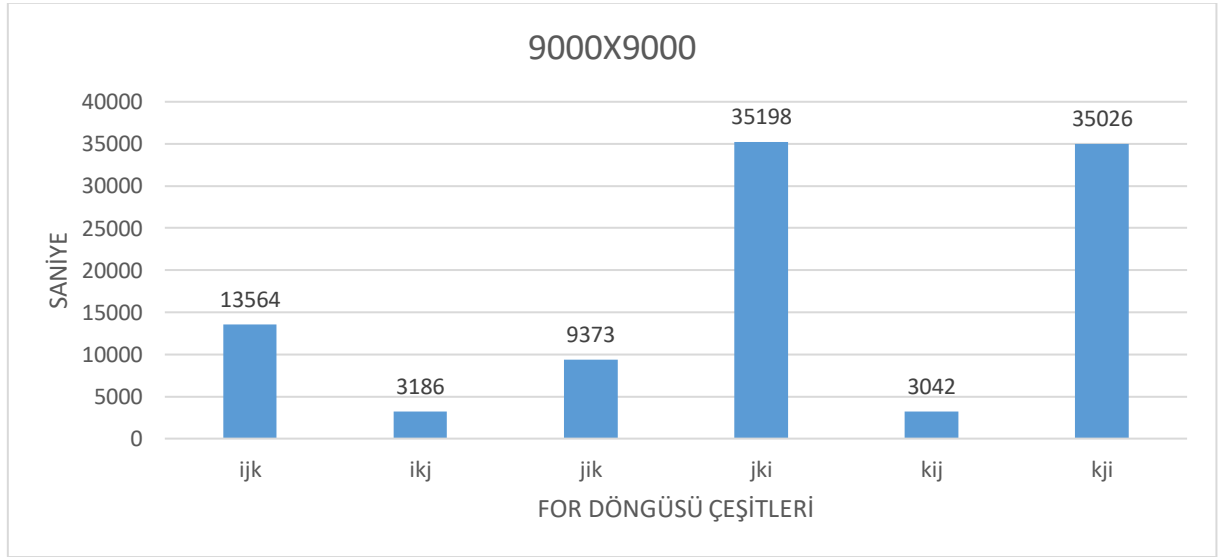
TÜM FOR DÖNGÜLERİ İÇİN 3000X3000 ZAMAN GRAFİĞİ



TÜM FOR DÖNGÜLERİ İÇİN 6000X6000 ZAMAN GRAFİĞİ



TÜM FOR DÖNGÜLERİ İÇİN 9000X9000 ZAMAN GRAFİĞİ



YORUMLAMA

$i \rightarrow k \rightarrow j$ algoritması ile en yavaş çalışan $k \rightarrow j \rightarrow i$ algoritması arasında 10 kat hız farkı var. Genel olarak matris çarpımı için kullanılan $i \rightarrow j \rightarrow k$ algoritmasından ise $i \rightarrow k \rightarrow j$ algoritması 4 kat daha hızlı çalışmaktadır. Algoritmalar arasında oluşan bu kadar büyük hız farklarının nedeni algoritmalar arasındaki önbellek kullanım farkından kaynaklanmaktadır.

Matris çarpma işlemlerinin gerçekleştirildiği iç içe 3 for döngüsü değişkenlerinin kullanım sırasına göre i,k,j ve k,i,j şeklinde uygulanan algoritmalarının çalışma mantığına bakıldığında, 1. ve 2. matrisin sahip olduğu elemanlarının tekrar tekrar gereksizce önbelleğe getirilmediği görüldü. matrix1 matrisindeki bir eleman ve matrix2 matrisindeki bir satırın işi bitinceye kadar önbellekte kalmakta ve işi bittikten sonra önbellekten silinmektedir.

Çarpma işleminin yapıldığı iç içe 3 for döngü değişkenlerinin sırasıyla i,j,k ve j,i,k şeklinde uygulanan algoritmalarda matrix2 matrisinin her bir sütununun matrix1 matrisinin her satırı için bir verinin görevi tamamlanmadan tekrar tekrar ön belleğe getirildiği görülmektedir. En içteki for döngüsünde matrix1 matrisinin sütunları ile matrix2 matrisinin satır elemanları çarpılarak matrix3 matrisin bir elemanı tamamen elde edilmektedir. Buradaki i,j,k algoritması insanların standart olarak matris çarpma işlemlerinde uyguladıkları yöntemi temsil etmektedir. Bir dıştaki yani ortadaki for döngüsü hesaplama süresince matrix2 matrisinin ön bellekteki elemanları silinerek sıradaki sütun elemanlarını ön belleğe almaktadır. Bu hesaplama algoritmalarına göre en dıştaki for döngüsü çalıştığı sürece matrix1 matrisinin her satırı için matrix2 matrisinin her sütunu belleğe tekrar tekrar alınmaktadır. Bunun sonucunda bu iki algoritma i,k,j ve k,i,j algoritmalarına göre daha yavaş çalışmaktadır. Çünkü fazladan ön belleğe taşıma ve silme işlemi gerçekleştirilmektedir.

Son olarak k,j,i ve j,k,i döngü değişkenlerinin sıralamasına göre oluşturulan algoritmalarda ön belleğe veri taşıma işlemi en verimsiz şekilde gerçekleştirilmiş, iş yükü ve veri trafiği artırılmıştır. Sürekli ön belleğe taşıma ve silme işlemi gerçekleştirilmiş ve bir verinin görevi bitmeden bir başka veri ön belleğe getirilip çalıştırılmıştır. Daha sonra tekrar görevi bitmeyen veri ön belleğe getirilmiş işlemlere bu döngüde devam edilmiştir. Bu nedenle en verimsiz algoritmalar bunlardır.

Kodlardaki en içteki for döngüsünde matrix1 matrisinin 1. elemanı matrix2 matrisinin 1. satır elemanlarıyla çarpılarak sonuç matrisinin (matrix3) 1. satırının bir bölümünü hesaplanmaktadır. Ancak sonuç matrisinin ilgili kısmı tam olarak tamamlanmamaktadır. Her adımda bir kısmı tamamlanmakta ayrıca ortadaki for döngüsünde her bir değer değiştiğinde ön bellekte yalnızca bir matrisin aynı sütundaki tek elemanı önbelleğe getirilmektedir. Yani sadece bir eleman değişikliği olmaktadır.

k,i,j ve i,k,j sıralamasına göre oluşturulan algoritmaların arasındaki fark ise, matrix1 matrisi ile matrix2 matrisinin yer değiştirmiş şekli olmasıdır. Yani cache belleğe taşınan verilerin birbirine yakın yöntemle taşınmasından kaynaklanmaktadır. Bu nedenle yakın sonuçlar elde edilmiştir.

Yapılan bu hesaplamalarda cache belleğe getirilen bir verinin daha sonra tekrar tekrar Ram'den ön belleğe taşınmadan ya da olabildiğince daha az miktarda taşıma işlemi gerçekleştirilerek hesaplamaların tamamlanmasını sağlamak amaçlanmıştır. Matris işlemlerinde alışıla gelen ($i \rightarrow j \rightarrow k$) çarpma işlemi yerine görünüşte insanlar için daha karmaşık olabilecek farklı mantıkta çarpma işleminin gerçekleştirilmesi ile cache bellek daha etkin kullanılmış ve işlem süresine olan olumlu ve olumsuz etkilerinin nasıl değiştiği gözlemlenmiştir.

BİLGİSAYAR ÖZELİKLERİ

CPU

Intel Core i5 @ 2.80GHz Haswell 22nm Technology

RAM Memory Size

12172 MBytes

Caches

L1 Data Cache Size 2 x 32 KBytes

L1 Instructions Cache Size 2 x 32 KBytes

L2 Unified Cache Size 2 x 256 KBytes

L3 Unified Cache Size 3072 KBytes

Operating System

Windows 10 Education 64-bit

KAYNAKÇA

M. Keskenler ve E. Keskenler , "Yoğun İşlem Yüküne Sahip Matris Çarpımı Hesaplama Sürelerinin Önbellek Kullanım Optimizasyonu ve Paralel Programlama Teknikleri Kullanılarak İyileştirilmesi", *Muş Alparslan Üniversitesi Fen Bilimleri Dergisi*, c. 6, sayı. 2, ss. 545-551, Ara. 2018