

Fatih Sultan Mehmet Vakıf Üniversitesi

Bilgisayar Mühendisliği

Hazırlayan Mevlüt Emre TANRIVERDİ

Okul NO 1821221023

Ders Bilgisayar Programlama III

Konu LAB - Ödev 1
(Matris – Matris çarpımı)



Amaç

Bu ödevin amacı C programlama dilindeki çok boyutlu dizilerin kullanımı ve bellek yönetimi kavramını öğrenmektir. Bunların yanı sıra C’de belleğe erişimin programın çalışma süresine etkisi de irdelenmiş olacaktır.

Ölçümlerin Gerçekleştirildiği Bilgisayara Ait Teknik Özellikleri



Yöntem

İlk olarak:

```
float **carpimMatris= matrisAlanOlustur(x);
```

```
float **matrisA = matrisAlanOlustur(x);
```

```
float **matrisB = matrisAlanOlustur(x);
```

İle iki boyutlu olmak üzere üç adet matrise yer açılmıştır.

```
matrisAlanDoldur(matrisA,x);
```

```
matrisAlanDoldur(matrisB,x);
```

fonksiyonları ile matrislere scanf ile alınan ölçütte random sayılarla iki boyuya doldurulmaktadır.

Devamında,

```
void **ijkmatrixMatrixMultiply(float **matrisA , float **matrisB, float **carpimMatris, int olcek);
```

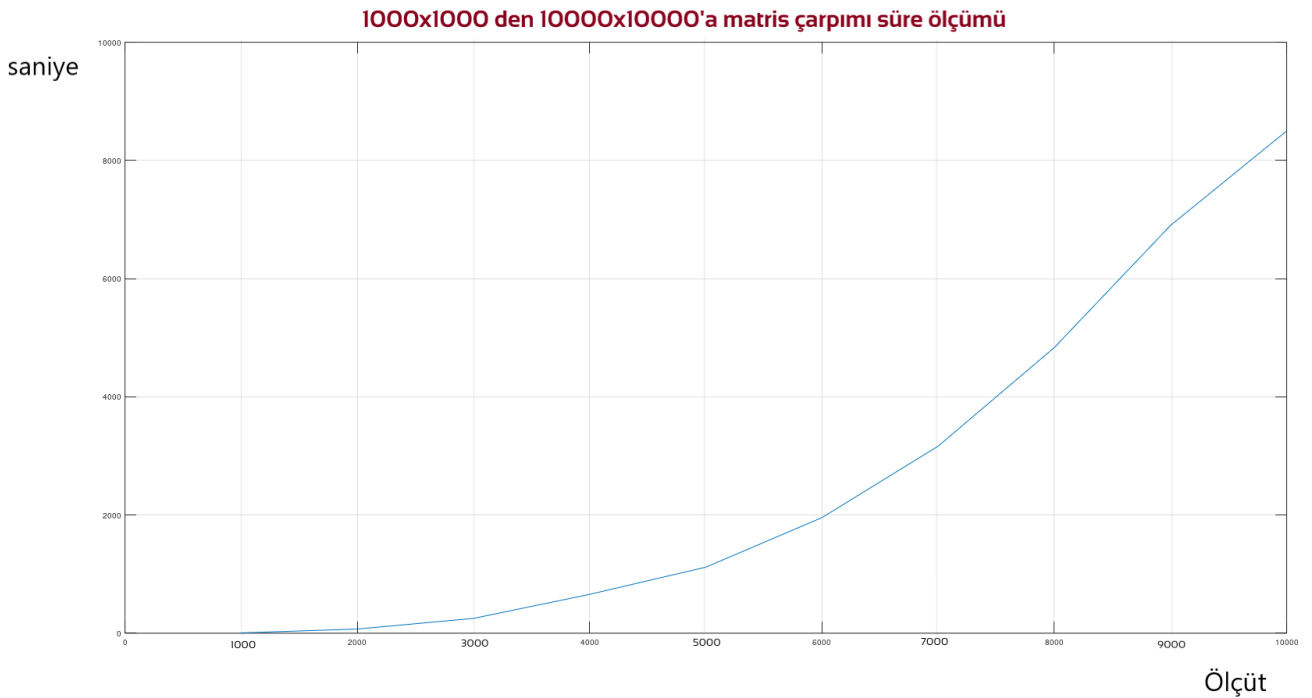
fonskyionu ile istenen döngü biçiminde matrisler çarpılmaktadır.

En son olarakta:

```
start = clock();  
  
    ijkmatrixMatrixMultiply(matrisA,matrisB,carpimMatris,x);  
  
end = clock();  
  
cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC;  
  
printf(" i j k dongu suresi : %f",cpu_time_used );  
  
puts("\n");
```

ile çarpılan döngülerin sürelerini hesaplanmaktadır.

Grafikler



Bu grafikte deęerlerini random olarak rastgele belirlediđimiz ve **i-j-k** ds dahilinde arptıđımız matrislerin arpım sonularını grmekteyiz. 1000x1000 den bařlayarak 10000-10000 ltleriyle olarak arptıđımız matrisimiz řu sonuları almıřtır:

1000x1000: 6.211 saniye

2000x2000: 71.786 saniye

3000x3000: 251.566 saniye

4000x4000: 658.983 saniye

5000x5000: 1118.815 saniye

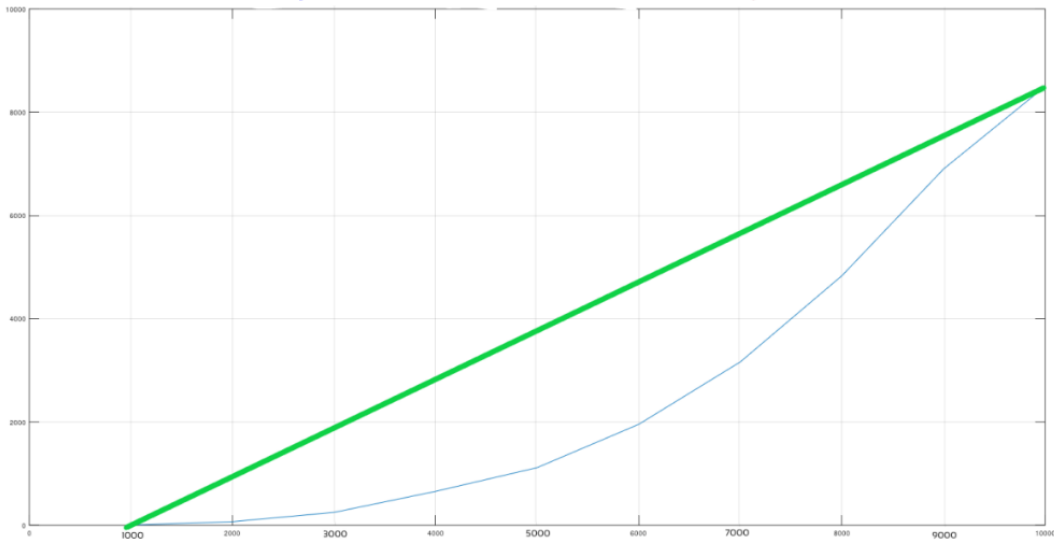
6000x6000: 1976.514 saniye

7000x7000: 3166.865 saniye

8000x8000: 4835.530 saniye

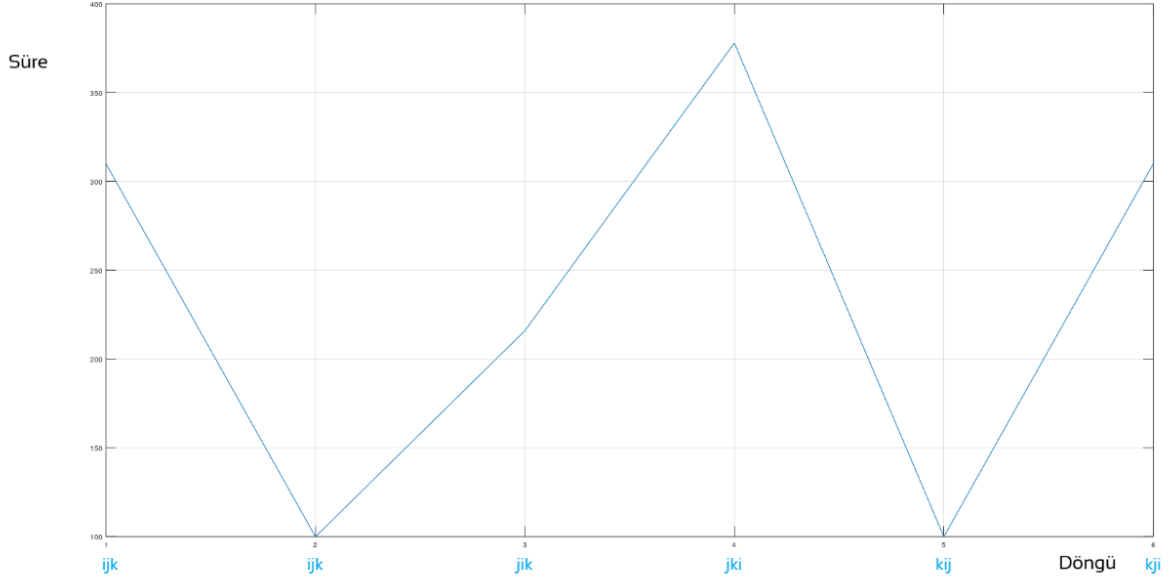
9000x9000: 6901.621 saniye

10000x10000: 8024.146 saniye

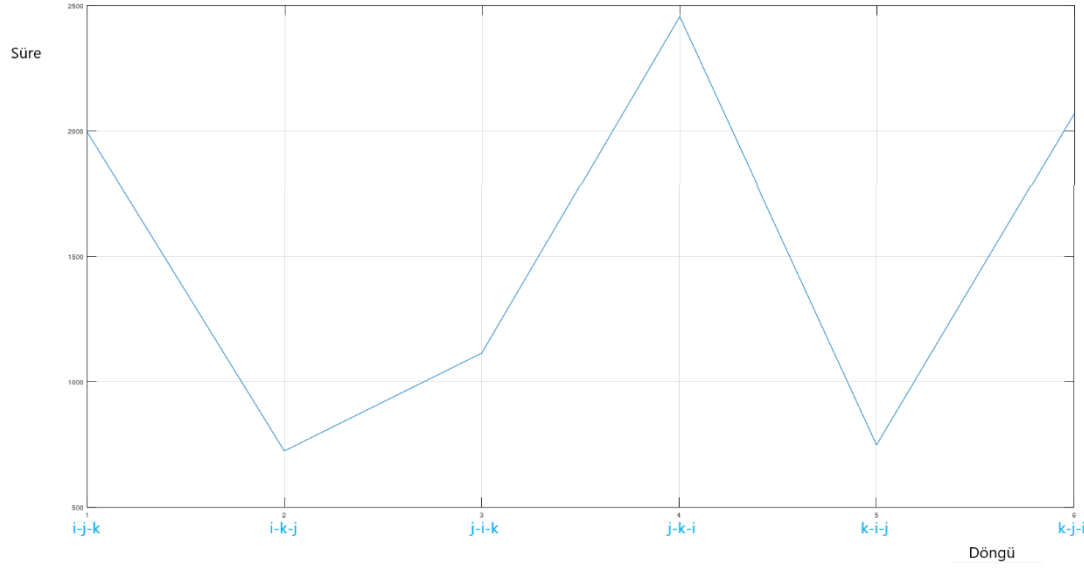


Sonulara bakıldıđında da ijk ds lt/sre hususunda arpılan ltlerin boyutlarında arttırıldıđında sre lmnde ivme kazanarak hızlanan trigonometrik bir artıř gzlemlenmektedir. zellikle 6000x6000 ltlerinden sonra matris arpım sreleri epey bymektedir.

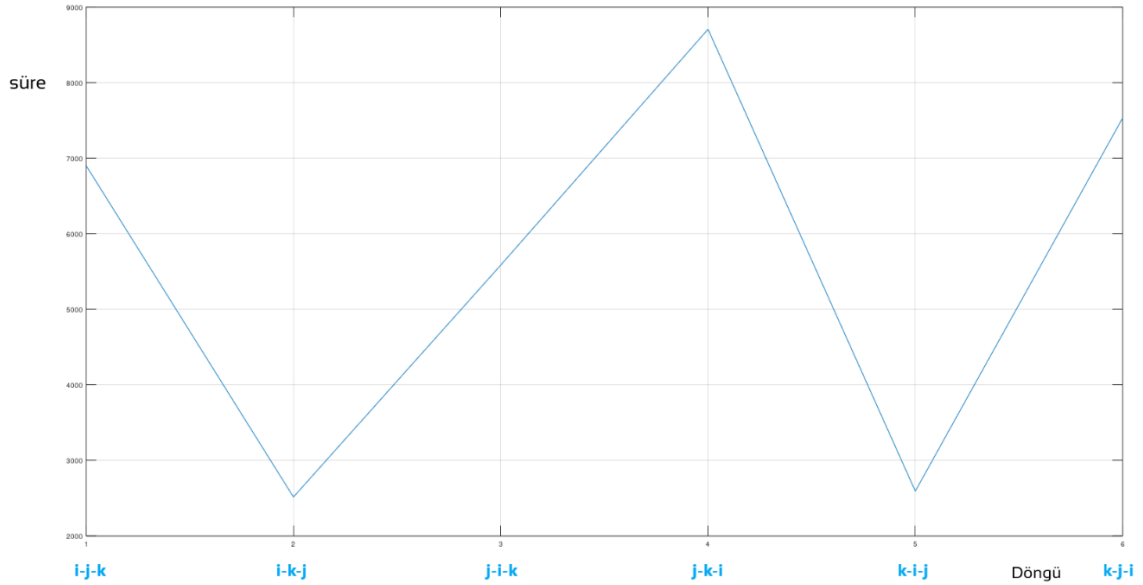
3000x3000 I-j-k, i-k-j, j-i-k, j-k-i, k-i-j ve k-j-i Döngülerinde Süre Ölçümü



6000X6000 i-j-k,i-k-j, j-i-k, j-k-i, k-i-j ve k-j-i Döngülerinde Süre Ölçümü



9000x9000 I-j-k, i-k-j, j-i-k, j-k-i, k-i-j ve k-j-i Döngülerinde Süre Ölçümü



Yorumlar

Başta döngü olarak i-j-k olarak baz aldığımız çarpımlarımızı i-k-j, j-i-k, j-k-i, k-i-j ve k-j-i döngüleriyle devam ettirilmiştir. Ve bu döngüleri:

3000x3000 6000x6000 ve 9000x9000 olarak çarpılmıştır ve sonuçlar süre/grafik olmak üzere kaydedilmiştir. Grafiklere bakıldığında :

Matris çarparken veriler bellekten ön belleğe taşınır. Devamında işlemci ön bellek üzerinde işlemleri tamamlar ve çıkan sonucu tekrar RAM e gönderir. Ön belleğin kapasitesi kısıtlı olduğu için matrisler yüksek boyutlarda çarpımı yapıldığında verilerin hepsi ön belleğe sığmaz ve ön bellek ile bellek arasında sürekli işlenecek veriler getirilip götürülür.

Matrislerin ölçümleri kaydedilirken i,k,j ve k,i,j şeklinde uygulanan algoritmalarının çalışma mantığına bakıldığında A ve B matrislerini gereksizce ön belleğe getirmedeği gözlemlendi. Buda sürelerin daha düşük çıkmasına neden oldu. **Ancak hepimizin standart olarak kullandığı i,j,k döngü sırası bu ikisine nazaran daha yavaş olduğu görülmektedir. Çünkü matris A nın her satırı için matris B ön belleğe çağrılmak zorunda kalmıştır.**

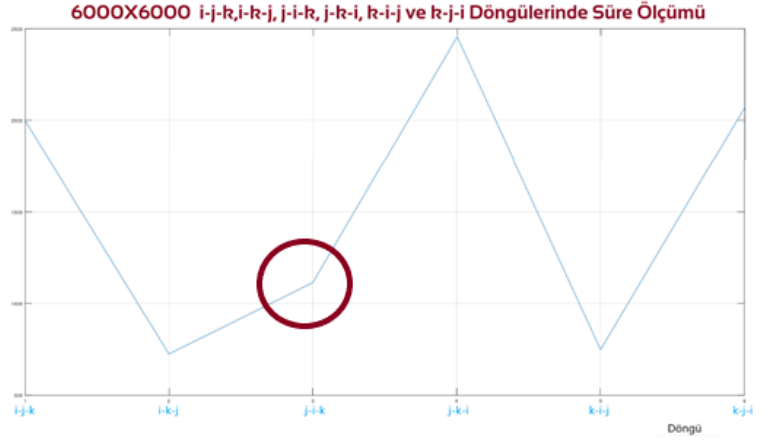
K,J,i ve J,K,i döngülerine baktığımızda bunların en verimsiz olduğu anlaşılmaktadır. Bu işlemlere baktığımızda sürekli ön belleğe taşıma ve silme işlemi gerçekleşmektedir. Ve görevler bitmeden ön belleğe veri getirilip silinmektedir. Bu durum bu iki döngüyü çok verimsiz kılmaktadır.

1.) **Her bir ölçütün W şekline olduğu eğri çizdiği gözlenmektedir.** Buna sebep olarak her bir ölçümün yapıldığı sıralama aynı olmaktadır. Ve çarpım indisleri 1000 er olarak düzenli arttırılmaktadır.

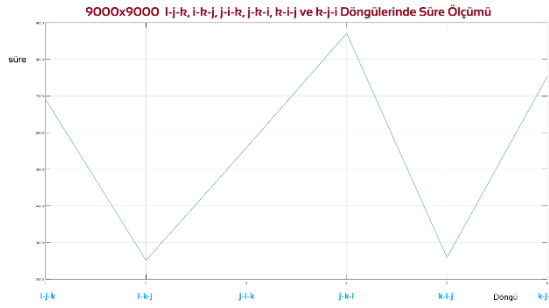
2.) **J-K-i döngüsü içerisinde çarpımların en uzun sürelerle neden olmaktadır.** Bunun sebebi olarak JKİ matrisi çarpım için döngüde yolu uzatmaktadır. Ve uzayan yol ölçüm sürelerini de uzatmaktadır

3.) 6000x6000 döngüsünün olduğu grafiğin bir kısmında eğride yamulmalar olmaktadır.

Buna neden olan o kısımda bilgisayarın gücünün pile takılmasıdır ve bu sebepten ötürü döngü hızlanmıştır



4.) 9000x9000 döngüsü en düze yakın döngüdür. Bunun sebebi 9000x9000 döngüsünde bilgisayar yaklaşık 10 saat çarpım dışında bir amaç için kullanılmamıştır. Bu sayede işlemci sadece çarpmakla ilgilenmiş ve aralara başka işlemler girmemiştir. Dolayısıyla eğri düze yakın sonuç vermiştir.



Sonuç

Matris çarpımına süre kazancı için en ideal döngü sırası İ,K,J VE K,İ,J dir.