



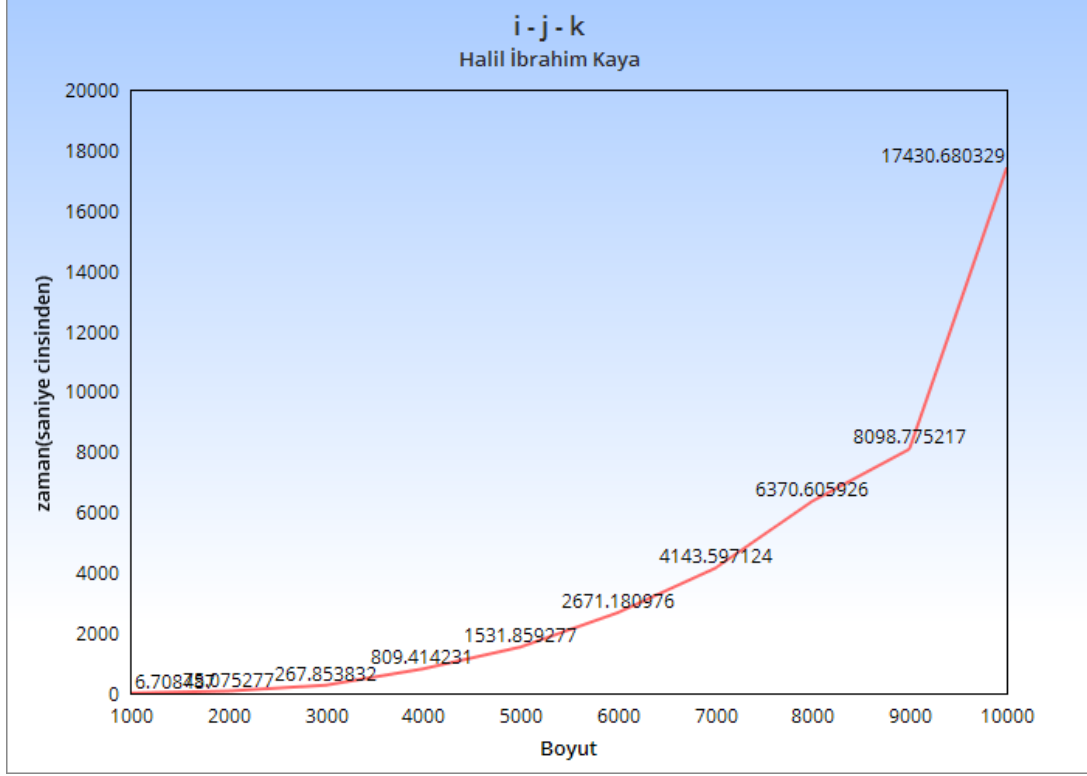
Bilgisayar Programlama III - Ödev I

Ad-Soyad:	Halil İbrahim Kaya
No:	172 122 1017
Ders:	Bilgisayar Programlama III
Ödev:	Ödev I

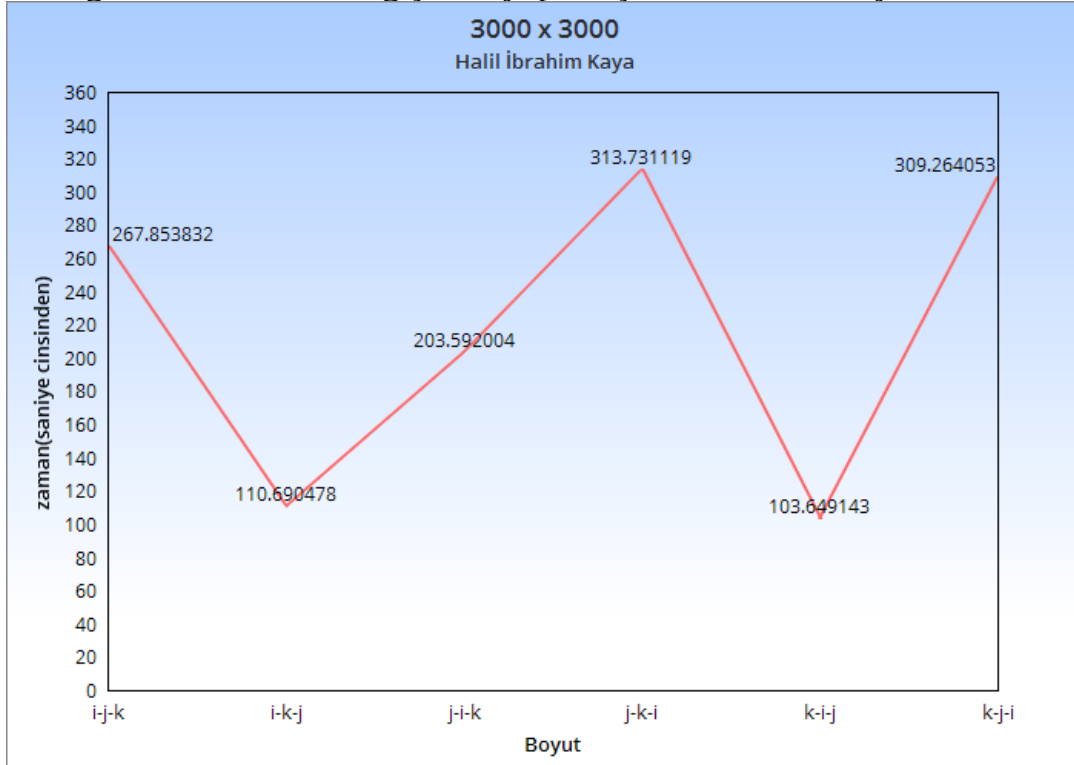
İşletim sistemi:	Linux Ubuntu
Ram:	24Gb
İşlemci:	i7-7500U

Grafikler:

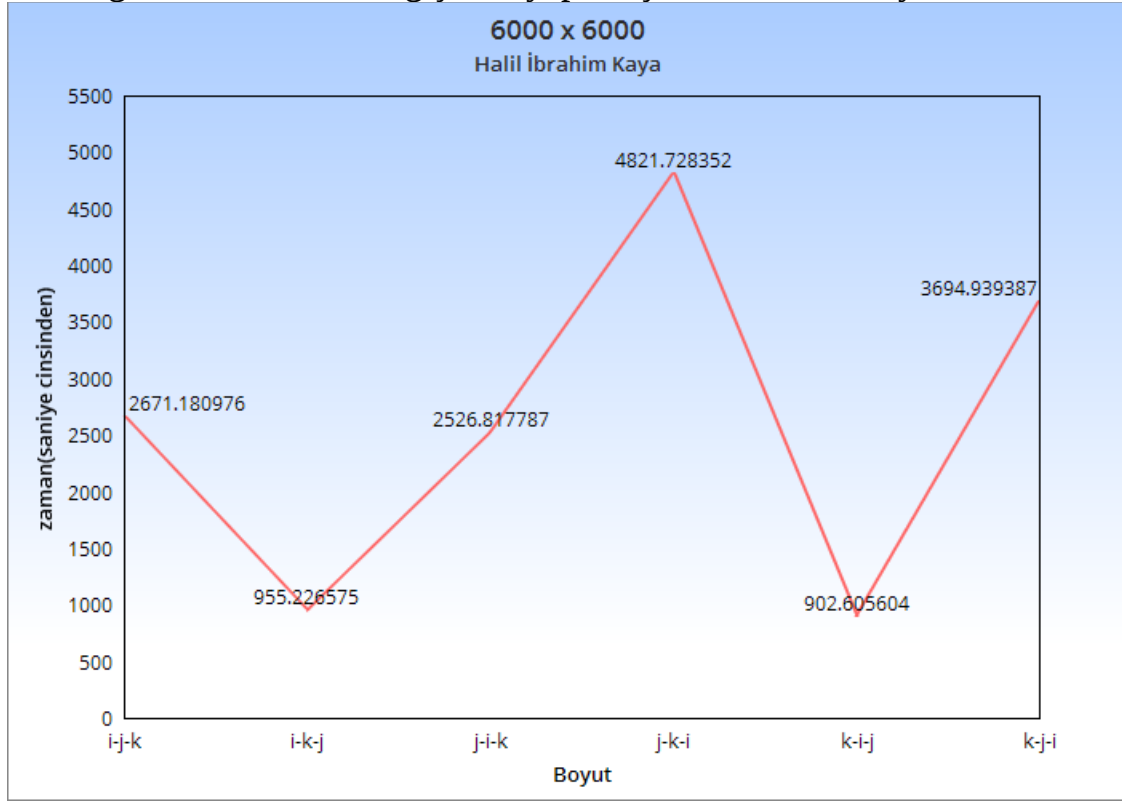
[Döngü sıralamasında değişiklik yapılmamış halinin sonucu]



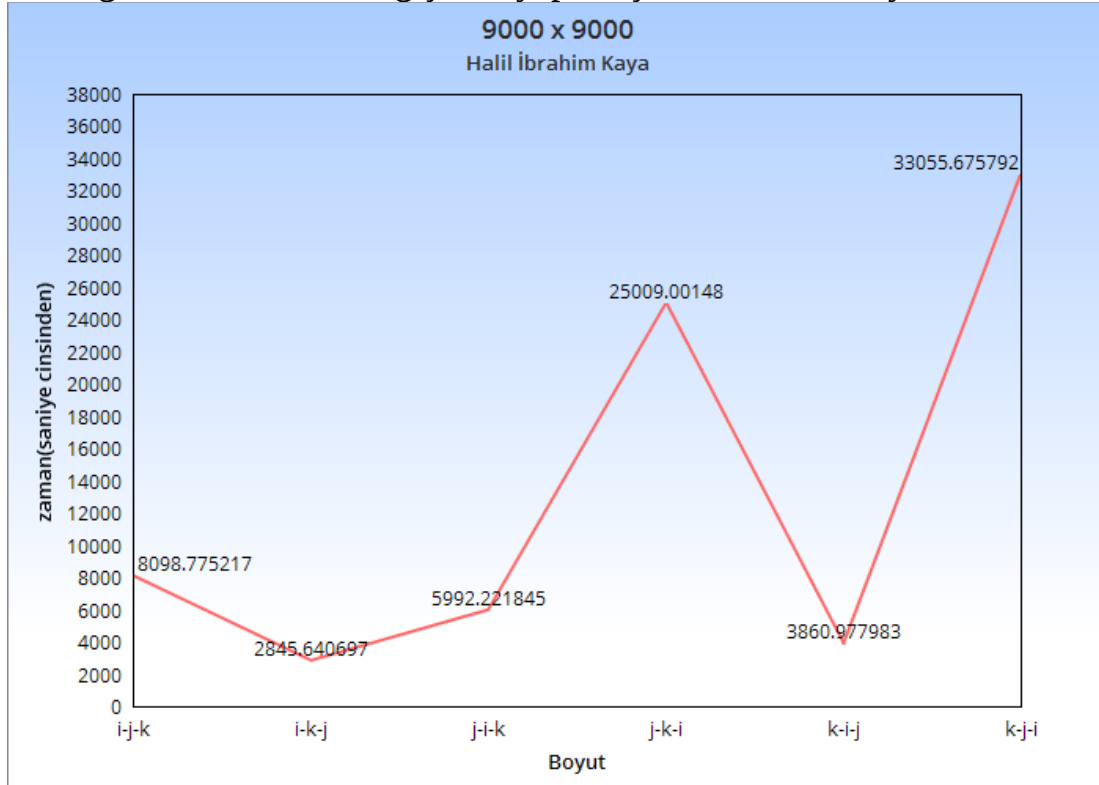
[Döngü sıralamasında değişiklik yapılmış halinin 3000 boyutlu sonucu]



[Döngü sıralamasında deęişiklik yapılmıř halinin 6000 boyutlu sonucu]



[Döngü sıralamasında deęişiklik yapılmıř halinin 9000 boyutlu sonucu]



[Kullanılan Algoritma]

```
for(i = 0 ; i < boyut ; i++){  
    for(j = 0 ; j < boyut ; j++){  
        for(k = 0 ; k < boyut ; k++){  
            (*(sonuc+i)+j) += ( (*(matrix1 + i)+k) ) * ( (*(matrix2 + k) +j) );  
        }  
    }  
}
```

Sebebi:

[internetten bulunan sonuç]

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, matrix1 matrisi ve matrix2 matrisinin sahip olduğu elemanlarının tekrar tekrar gereksizce ön belleğe getirilmediği görüldü. matrix1 matrisindeki bir eleman ve matrix2 matrisindeki bir satırın işi bitinceye kadar ön bellekte kalmakta ve işi bittikten sonra ön bellekten 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 sonuc 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,

[yorum]

Ödevden çıkarılan ders ve edinilen tecrübe şu; Ramle işlemci arasındaki ilişkiyi bilmek ve onun nasıl çalıştığını anlamak bize algoritmaların daha performanslı çalışmasına olanak sağlayabilir bu ödevden de gördüğümüz gibi bir kaç değişkenin yerini değiştirmemiz bize problem çözümünde saatler kazandırdı. İşlemlerin bu kadar uzun sürmesinin sebebine örnek olarak matrix1 deki bir elemanın matrix2 deki bir satırın işi bitinceye kadar belleğe tekrar ve tekrar alınmasını gösterebiliriz.