

2020 – 2021 GÜZ Bilgisayar Programlama III ve Bilgisayar Programlama III LAB - Ödev 1

(Matris – Matris çarpımı)

AD: ADEM YAVUZ ÇELİK

NUMARA: 1521221055

İŞLEMCI: 64 bitlik 3.5 GHz

L1=384 KB ,

L2 = 2 MB,

L3=32MB

RAM: (2*8)gb 3200 Mhz RAM

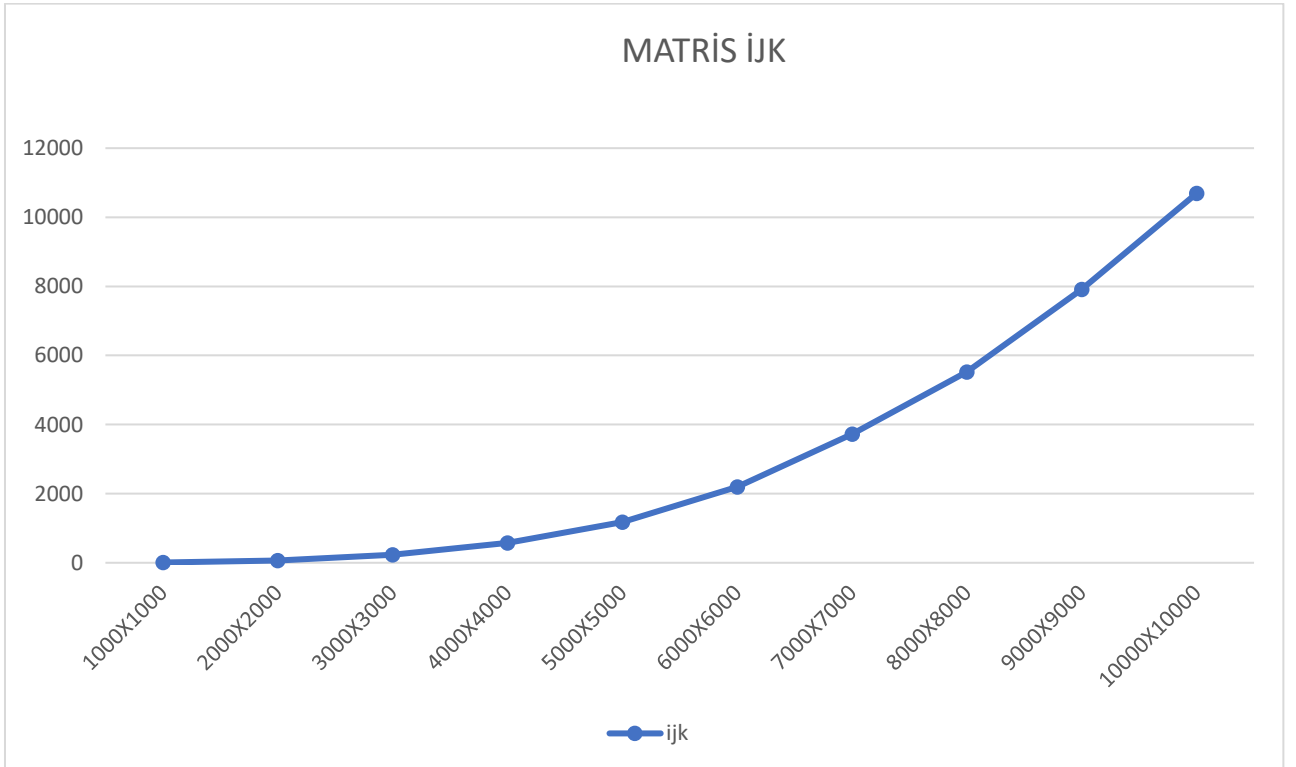
İŞLETİM SİSTEMİ:WINDOWS

1.madde: i-j-k sırası için yatay eksende matris boyutu, dikey eksende süre (saniye) olmak üzere bir adet grafik çizilecektir.

DEĞERLER

MATRİS BOYUT	SANİYE
1000X1000	8,4
2000X2000	68,37
3000X3000	236
4000X4000	578,2
5000X5000	1180
6000X6000	2201
7000X7000	3728
8000X8000	5524
9000X9000	7910
10000X10000	1.069E+004->10690

GRAFİK

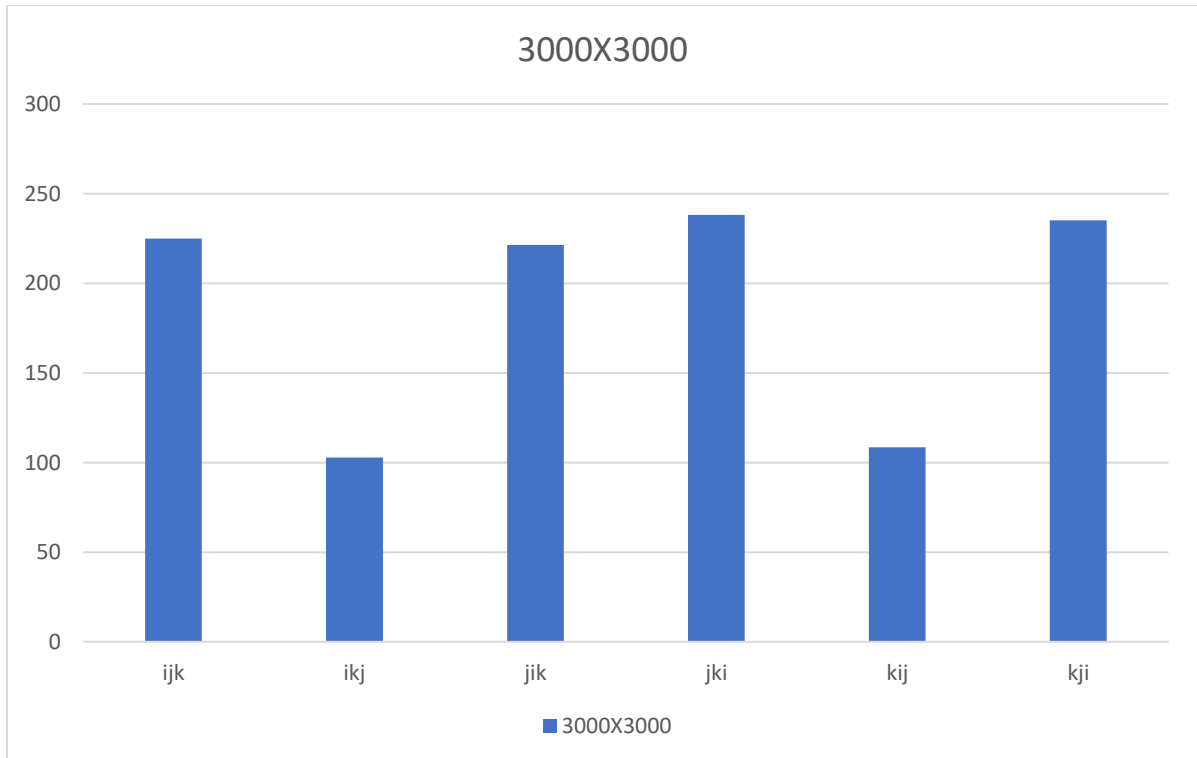


2.madde:3000 × 3000, 6000 × 6000 ve 9000 × 9000 boyutları için yukarıda belirtilen 6 sıranın 6'sı için de süre ölçümü alınacak ve 3 adet grafikte ayrı ayrı raporlanacaktır.

3000X3000 MATRİS DEĞERLER

DÖNGÜ DEĞİŞKENLERİ	SANİYE
İJK	225
İKJ	102,7
JİK	221,4
JKİ	238,2
KİJ	108,4
KJİ	235,1

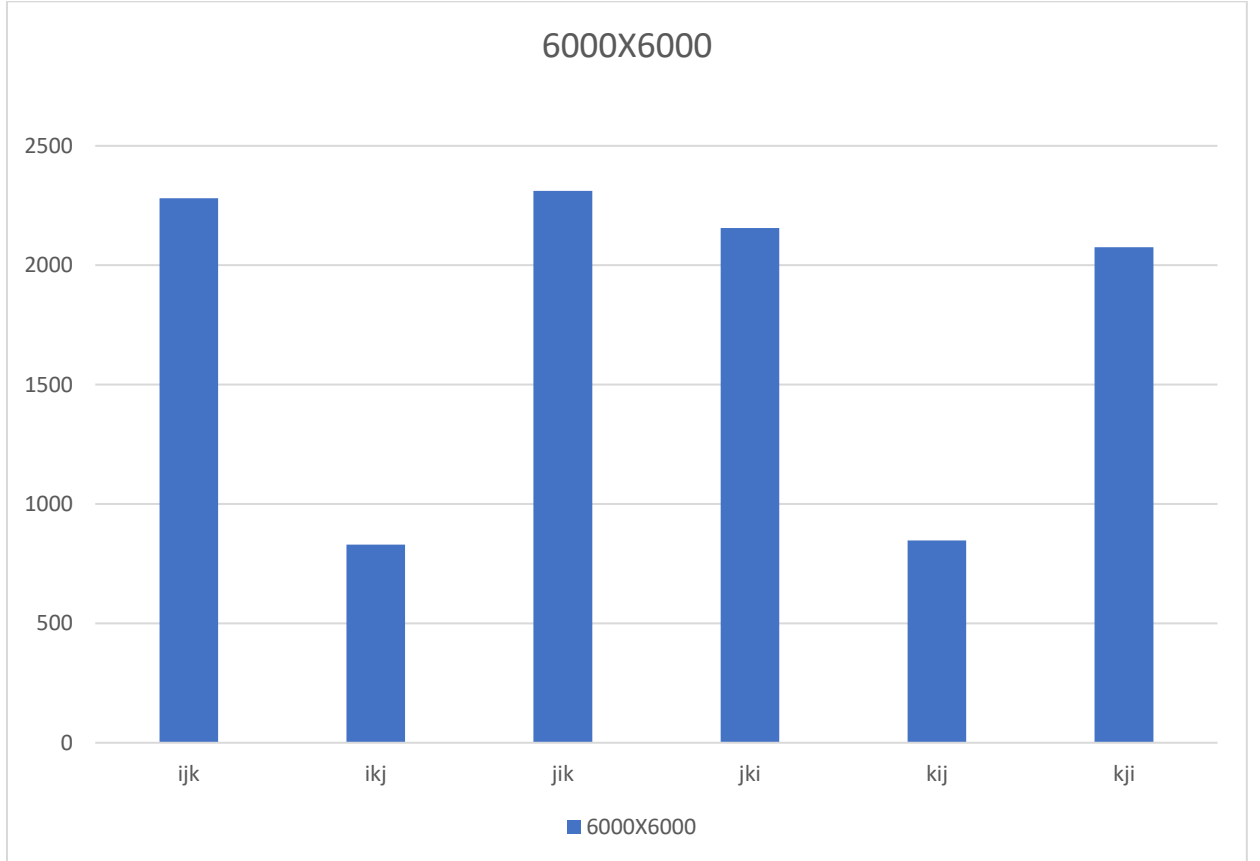
3000X3000 MATRİS GRAFİK



6000X6000 MATRİS DEĞERLER

DÖNGÜ DEĞİŞKENLERİ	SANİYE
İJK	2281
İKJ	829,7
JİK	2311
JKİ	2156
KİJ	846,7
KJİ	2076

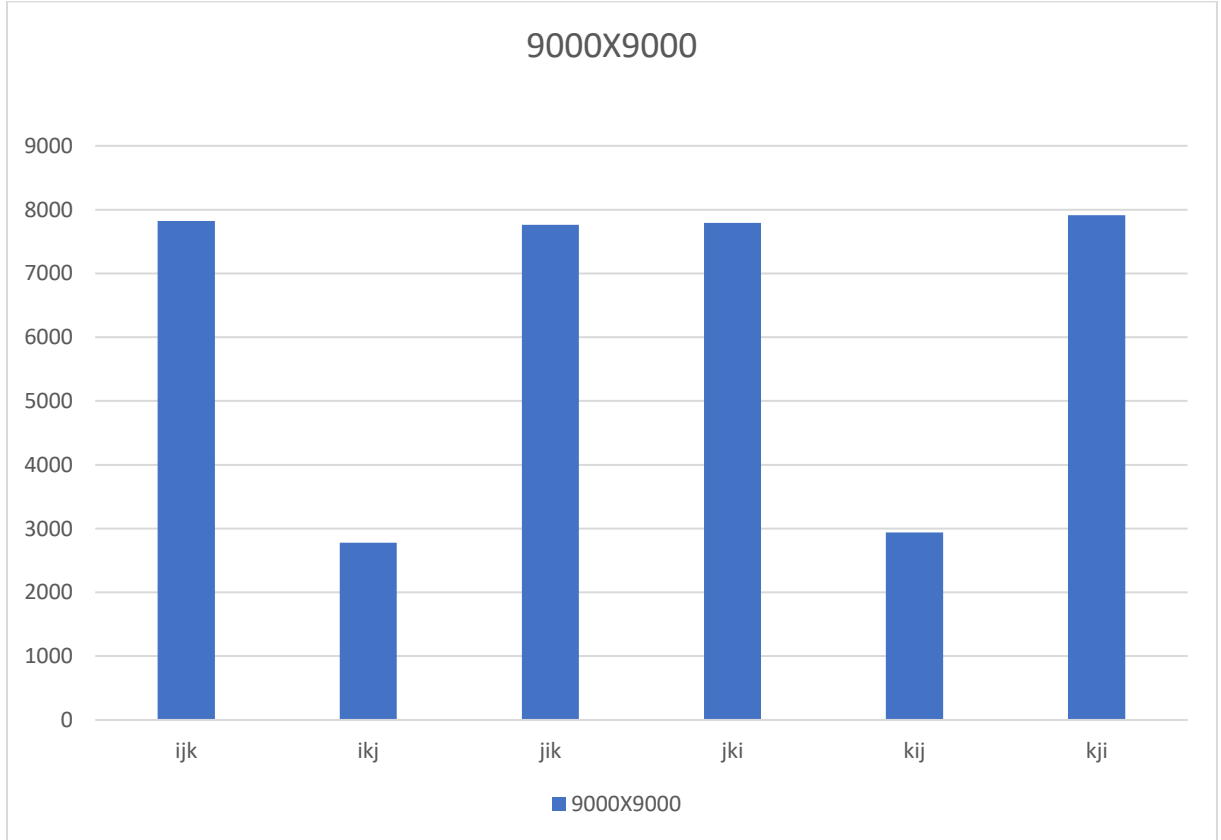
6000X6000 GRAFİK



9000X9000 MATRİS DEĞERLER

DÖNGÜ DEĞİŞKENLERİ	SANİYE
İJK	7822
İKJ	2781
JİK	7761
JKİ	7791
KİJ	2937
KJİ	7915

9000X9000 GRAFİK



YORUM: Kodda iki matrisin çarpımı için iç içe 3 for döngüsü kullanıldı. Bu for döngülerinde değişken döngüleri şu şekilde i-j-k, i-k-j, j-i-k, j-k-i, k-i-j ve k-j-i atandı. Matris çarpımı için aynı işlemi yapan fakat döngü değişken sıraları farklı olan 6 adet fonksiyon tanımlandı. Fonksiyonlara 3 er adet matris ve kare matris oldukları için tek bir boyut değişkeni gönderildi. İlk matrisimiz için **ptr, ikinci matrisimiz için **ptr2 son olarak dönecek matris için **ptr3 pointer to pointerları kullanıldı En içte bulunan for döngüsü yazılan kod sayesinde her bir döngü adımında dönecek olan çarpım matrisinin bir elemanı hesaplandı. Boyut değişikliği ve döngü değişkenlerinin sırasından bazen zamandan tasarruf edilirken bazense zamandan kayıp verildi. Bunun sebebi veriler RAM' e gönderilir. Veriler RAM' den işlemci ön belleğine taşınır. İşlemci, hesaplamaları tamamlar ve sonucu tekrar RAM' e gönderir. Kısıtlı sayıda veri tutabilen ön belleğe yüksek boyutlarda veri girilemez bu yüzden sürekli RAM ile veri alışverişi yapar buda neden matris boyutu arttıkça geçen sürenin arttığını anlamamıza yetiyor. Öte yandan matris boyutu sabit tutulup sadece döngü değişkenlerinin sırası değiştirildiğinde görüldüğü üzere i-k-j ve k-i-j değişkenleri diğer 4 değişken sıralamasına göre ciddi bir zaman tasarrufu sağlıyor sebebi yine ön bellek kullanımından çünkü i-k-j ve k-i-j değişkenlerinde kodun çalışma mantığına bakılınca gereksiz yere ön belleğe tekrar tekrar verilerin gelmediği anlaşıyor, i-j-k,j-i-k, j-k-i, ve k-j-i de çarpılan iki matristeki elemanlar satır işlemi bitinceye kadar ön bellekte kalmakta ve iş bittikten sonra ön bellekten silinmektedir fakat bir işlem bitmeden bir sonraki veri ön belleğe getirilip iş yükü ve veri trafiği artırıldığından zamandan kayıp verilmektedir.