MULTI-THREADED MATRIX MULTIPLICATION

Satır ve sütun sayısı konsoldan girilen iki matris, konsoldan alınan thread sayısına göre , matrisin satırları threadlere bölünerek çarpma işlevini gerçekleştirir.

A ve B matrislerinin nxm boyutlarında çarpmayı gerçekleştirecek iki matris olduğunu ve bu çarpım sonucunun tutulacağı nxn boyutunda bir sonuç matrisinin olduğunu varsayarsak, nxn boyutundaki sonuç matrisinin threadler ile hesaplanması için A matrisinin n satırı threadlere bölünerek hesaplanmalı.

Matris çarpımının hesaplanmasının ne kadar sürdüğünü hesaplamak için long türünde başlangıç ve bitiş değerleri startTime ve endTime değişkenlerine atanıyor. Matrisin çarpma işlemini hesaplama süresi milisaniye cinsinden hesaplanır.

Matrisin satırı konsoldan aldığımız thread sayısı değerine bölünecek. Böylece girilen thread sayısı satır sayısına eşit değilse bir threadin hesaplaması gereken satır sayısını buluyoruz.

```
//başlangıç zamanı
long startTime = System.currentTimeMillis();

//Matrisin satırlarını girilen thread sayısına bölmek için
int step = rA / threadSayisi;
int startIndex = 0;
int endIndex = step;


//klavyeden girdi olarak alınan thread sayısı kadar thread üretmek için
for(int i = 0; i < threadSayisi; i++) {
    MatrisCarp carp = new MatrisCarp(matrisA, matrisB, sonuc, startIndex, endIndex, i);
    Thread thread = new Thread(carp);
    thread.start();

    startIndex = endIndex;

    if(i == threadSayisi-2) {
        endIndex = rA;
    }
    else {
        endIndex += step;
    }

    try {
       thread.join();
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
       e.printStackTrace();
    }
    MatrisYazdirma(sonuc);
}

System.out.println();
System.out.println("MATRİS SONUCU: ");
System.out.println();
MatrisYazdirma(sonuc);
```

```
//sona erme zamanı
long endTime = System.currentTimeMillis();

//Hesaplamanın tamamlanma süresi
System.out.println();
System.out.println("Çalışma süresi :  " + (endTime - startTime) + " milisaniye");
```

Matris çarpımının yapılacağı MatrisCarp sınıfı Runnable sınıfından implement ediliyor. MatrisCarp sınıfının constructoru, çarpımın yapılacağı A ve B matrislerini, sonucun tutulacağı S matrisini, threadi matrisin satır saysına böldüğümüz için başlangıç ve bitiş indislerini ve çalışan threadin hangi satırlarda işlem yaptığını göstermek için sayaç parametrelerini alıyor.

```
public class MatrisCarp implements Runnable{

	private int[][] A;
	private int[][] B;
	private int[][] S;
	private int startIndex;
	private int endIndex;
	private int sayac;

	public MatrisCarp(int[][] A, int[][] B, int[][] S, int startIndex, int endIndex,int
sayac) {

		this.A = A;
		this.B = B;
		this.S = S;
		this.startIndex = startIndex;
		this.endIndex = endIndex;
		this.sayac = sayac;
	}

	@Override
	public void run() {
		// TODO Auto-generated method stub

		System.out.println();
		System.out.println("Thread: " + (sayac + 1) + "  çalışıyor...");
		System.out.println();
		for(int i = startIndex; i < endIndex; i++) {
			for(int j = 0; j < B[0].length; j++) {
				S[i][j] = 0;
				for(int k = 0; k < A[i].length; k++) {

					S[i][j] += A[i][k] * B[k][j];
				}

			}
		}
	}

}
```

A ve B matrislerinin boyutları sırasıyla 7x6 ve 6x8 , thread sayısı 4 olarak klavyeden giriliyor. Boyutlar belirlendikten sonra, 0 ile 17 aralığındaki değerlerle matrisler rastgele oluşturuluyor. Daha sonra bu matrislerin tek tek threadlere çarptırılıp elde edilen sonuç ve her threade paylaştırılan satırların hesaplanma süreleri ekrana yazdırılır.

```
A Matrisinin Satır Sayısı: 7
A Matrisinin Sütun Sayısı: 6
B Matrisinin Satır Sayısı: 6
B Matrisinin Sütun Sayısı: 8

Thread sayısı giriniz: 4

A Matrisi:

2   0   3   12   7   1
16   7   7   4   8   12
16   7   11   5   6   10
9   13   12   14   9   10
13   0   5   3   7   1
7   15   13   11   9   7
2   8   10   9   8   15

B Matrisi:

14   1   13   4   8   4   10   13
5   2   1   13   10   11   13   1
15   12   2   1   2   5   3   15
9   15   4   13   14   13   10   4
2   1   14   11   6   3   15   15
2   1   3   7   2   14   0   0
```

```
Thread: 1  çalışıyor...
Matris Sonucu:

197   226   181   251   234   214   254   224
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0

Çalışma süresi :  3 milisaniye

Thread: 2  çalışıyor...
Matris Sonucu:

197   226   181   251   234   214   254   224
440   194   393   386   340   420   432   456
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0

Çalışma süresi :  1 milisaniye
```

```
Thread: 3  çalışıyor...
Matris Sonucu:

197   226   181   251   234   214   254   224
440   194   393   386   340   420   432   456
501   253   371   367   346   419   424   490
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0

Çalışma süresi :  3 milisaniye

Thread: 4  çalışıyor...
Matris Sonucu:

197   226   181   251   234   214   254   224
440   194   393   386   340   420   432   456
501   253   371   367   346   419   424   490
535   408   366   568   496   588   570   501
300   126   292   180   200   151   280   361
499   374   323   527   454   526   549   480
345   296   247   432   320   497   364   340

Çalışma süresi :  4 milisaniye
```