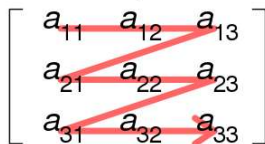
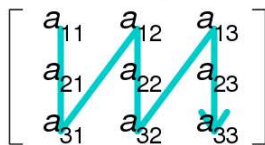


## השוואת זמני ריצה במעבר על שורות או עמודות ב-JAVA

### Row-major order



### Column-major order



נשאלת השאלה באיזה שיטה הקומפיילר של Java מבצע חיפוש מהיר יותר?

פריסת הנתונים היא קריטית להעברת מערכים נכונים בין תוכניות שנכתבו בשפות תכנות שונות. זה חשוב גם לביצועים בעת חציית מערך מכיוון שמעבדים מודרניים ה-CPU's רצופים ביעילות רבה יותר מאשר נתונים שאינם עוקבים. זה בעיקר בגלל מטמון של מעבד.

שפות תכנות שונות מטפלות בכך בדרכים שונות.

ב-C, מערכים רב-ממדיים נשמרים בסדר הראשי של השורה, ואינדקסי המערך נכתבים בשורה הראשונה.

```
final static int SIZE = 10000;
int[][] matrix = new int[SIZE][SIZE];
int counter = 0;
long start = System.currentTimeMillis();
for(int i = 0; i < matrix.length; i++) {
    for(int j = 0; j < matrix.length; j++) {
        matrix[i][j] = counter++;
    }
}
long end = System.currentTimeMillis();
long timeRow = end - start;
System.out.println("## TOTAL TIME FOR ROW ORDER: "+timeRow+"ms.");
counter = 0;
start = System.currentTimeMillis();
for(int i = 0; i < matrix.length; i++) {
    for(int j = 0; j < matrix.length; j++) {
        matrix[j][i] = counter++;
    }
}
end = System.currentTimeMillis();
long timeCol = end - start;
System.out.println("## TOTAL TIME FOR COLUMN ORDER: "+timeCol+"ms.");
```

בבדיקת סימולציה בשפת ג'אווה על זמני ריצה אפשר לראות באופן בולט כי הקומפיילר שלה עובד יעיל יותר בשיטת מעבר בין שורות ולא בעמודות:

```
#1 TOTAL COLUMN ORDER: 1216ms , TOTAL ROW ORDER: 47ms.
#2 TOTAL COLUMN ORDER: 1184ms , TOTAL ROW ORDER: 56ms.
#3 TOTAL COLUMN ORDER: 1175ms , TOTAL ROW ORDER: 48ms.
```