



Processamento da Informação

Exercícios sobre Matrizes

Prof. Jesús P. Mena-Chalco
CMCC/UFABC

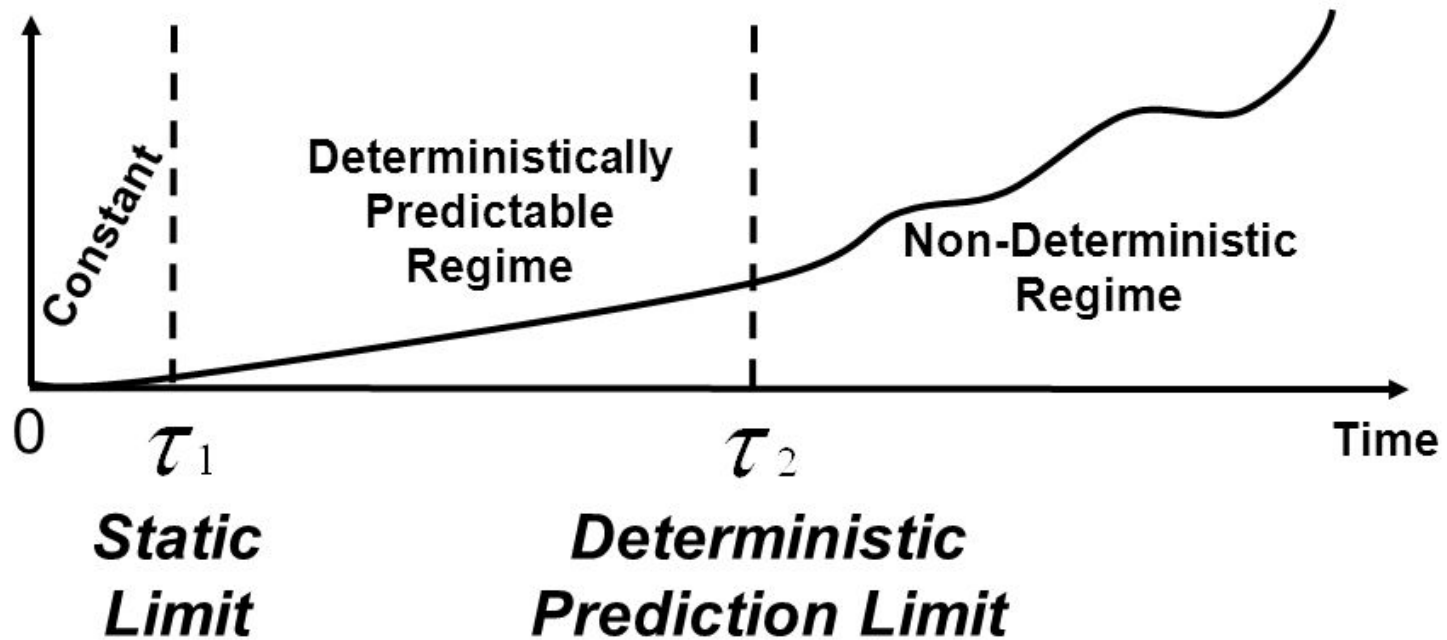
Q2/2018

Weather Time Scale Analysis

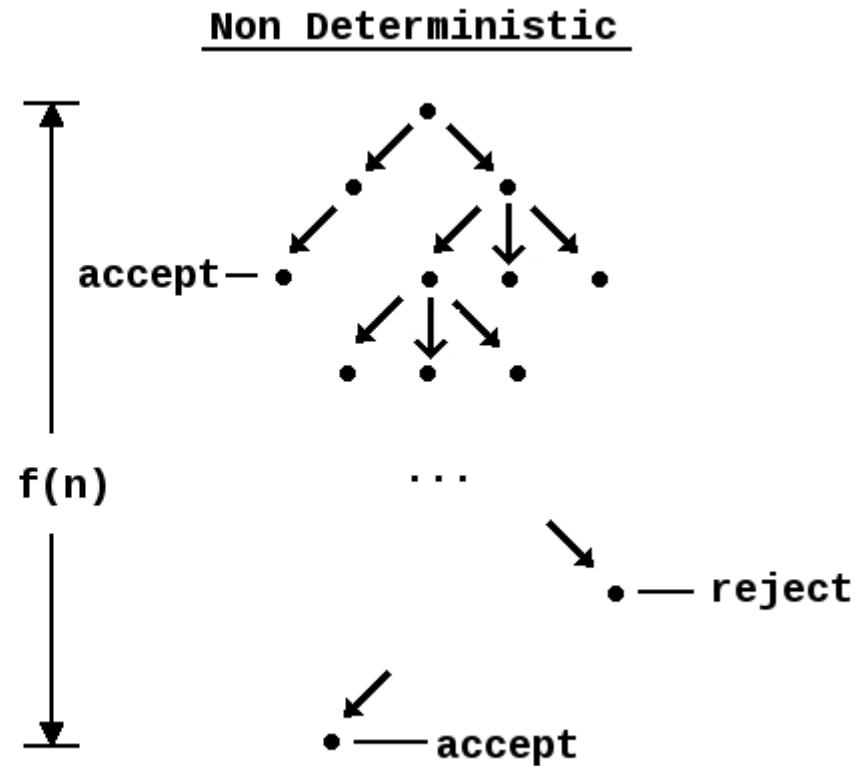
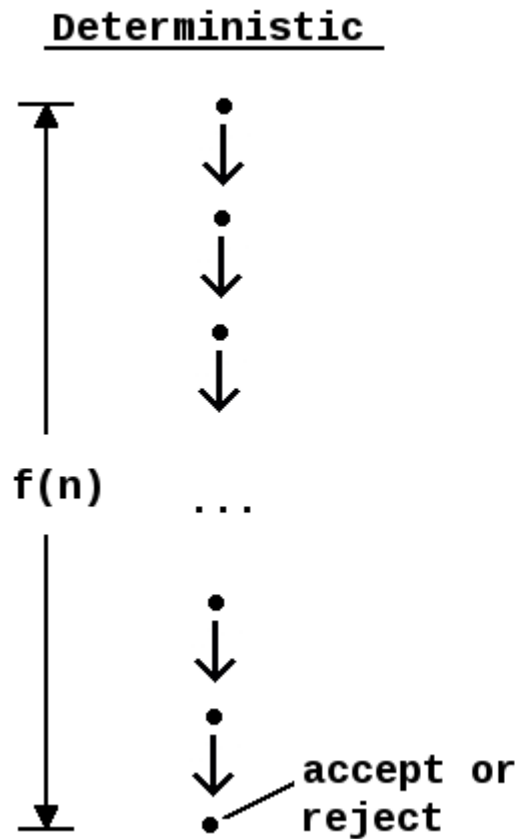
Concept



Error in
Phenomena
Knowledge



Algoritmos: Determinísticos Vs Não-determinísticos

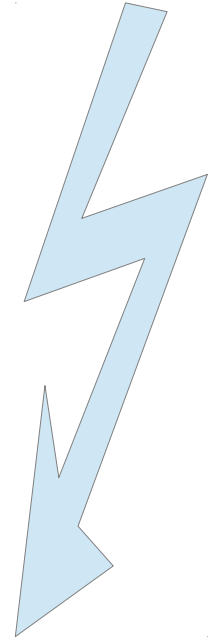


Um programa **determinístico** sempre gera a mesma saída para o mesmo conjunto de entrada.

Um programa é **não-determinístico** quando apresenta resultados diferentes com os mesmos conjuntos de entrada.

O que faz um algoritmo ser não-determinístico?

```
static int funcaoRA (int ra) {
    int aux, i=0, j;
    int v[] = new int[8];
    while (ra>0) {
        v[i] = ra%10;
        ra = ra/10;
        i = i+1;
    }
    for (i=1; i<8; i=i+1) {
        aux = v[i];
        for (j=i-1; j>=0 && v[j]>aux; j=j-1) {
            v[j+1] = v[j];
        }
        v[j+1] = aux;
    }
    return v[ v[7]%8 ];
}
```



- Erro em hardware.
- Opera de maneira sensível ao tempo.
- Usa valores aleatórios.

Não-determinístico é ruim?

As vezes é difícil criar algoritmos determinísticos que sejam “rápidos para responder” a uma entrada.

Existem algoritmos não-determinísticos que resolvem o problema com uma **ligeira margem de erro**.



Números Aleatórios

Números aleatórios

Sequência de números aleatórios são importantes em inúmeras aplicações.

É muito difícil obter números verdadeiramente aleatórios.

Geralmente os números são chamados de **pseudo-aleatórios** (pois são gerados por algoritmos)

```
int x = gerarNumeroAleatorio();
```

Números aleatórios

```
import java.math.*;
```

```
double x;  
int y;
```

```
x = Math.random(); // gera um numero real aleatorio [0, 1)
```

```
x = Math.random()*30; // gera um numero real aleatorio [0, 30)
```

```
y = (int) (Math.random()*30); // gera um numero inteiro aleatorio [0, 30)
```

```
y = (int) (Math.random()*30)+10; // gera um numero inteiro aleatorio [10, 40)
```

```
y = (int) (Math.random()*60)-30; // gera um numero inteiro aleatorio [-30, 30)
```



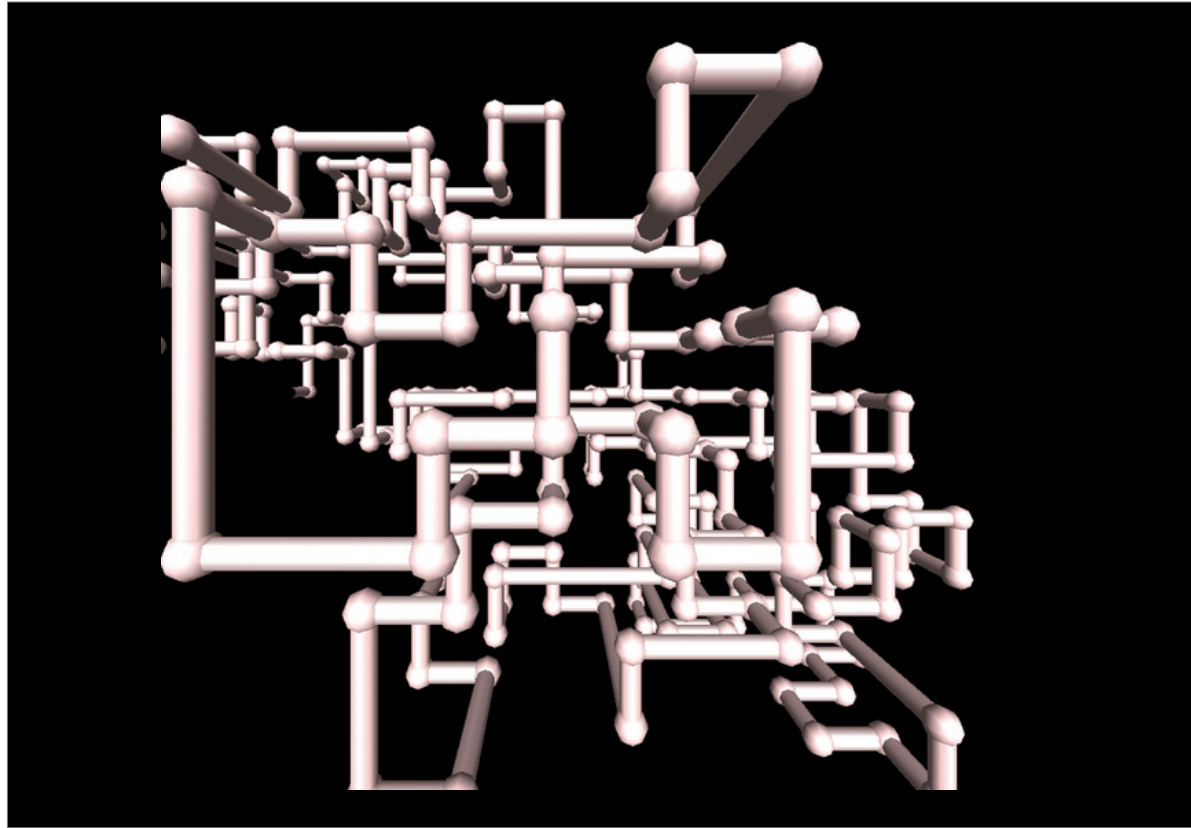

O que permite gerar a seguinte instrução?

```
(int)(Math.random()*1000)%10+1
```

Números aleatórios

→
Gerar um
número
aleatório

←
Pronto



Math.Randon

Números aleatórios

Semente (seed)

Temperatura

Hora atual do sistema

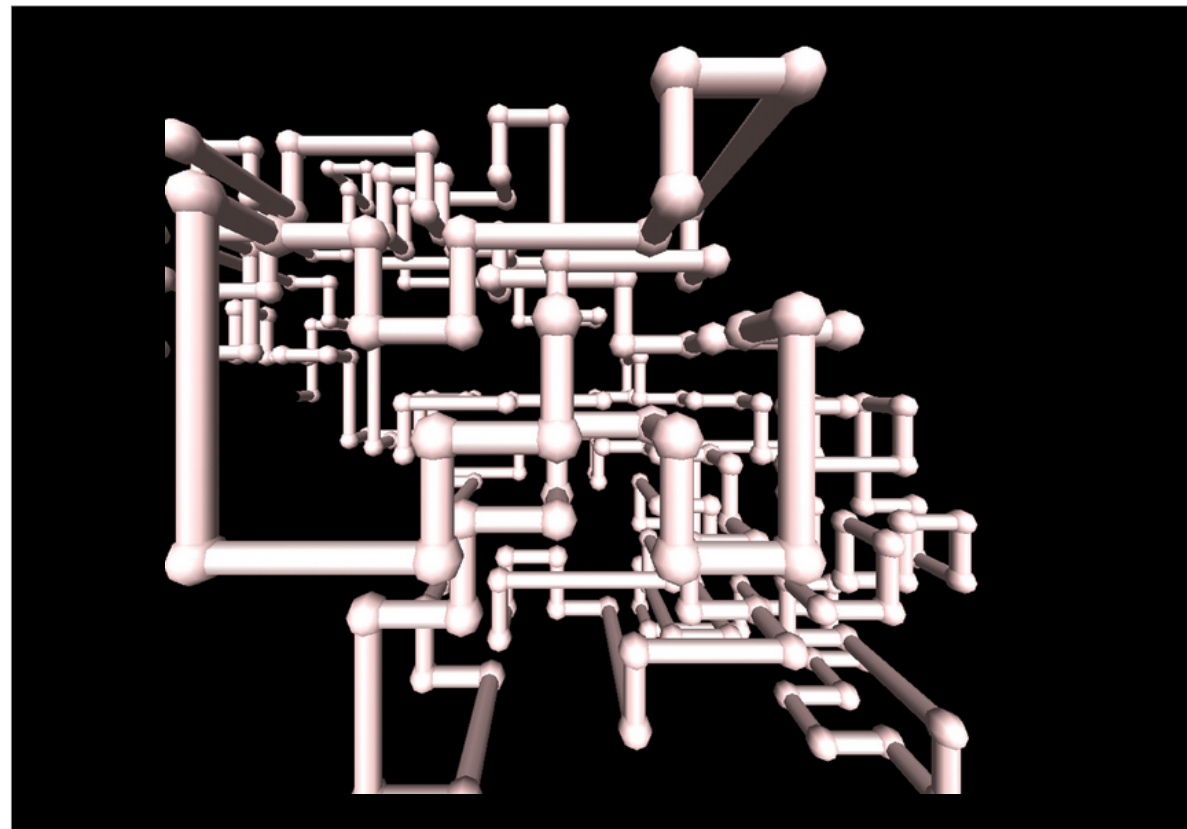
Número de rotações do HD

Material radiativo

...



Gerar um
número
aleatório



Pronto

Math.Randon

Números aleatórios

[Home](#) [Games](#) [Numbers](#) [Lists & More](#) [Drawings](#) [Web Tools](#) [Statistics](#) [Testimonials](#) [Learn More](#) [Login](#)

RANDOM.ORG

Search RANDOM.ORG
Google™ Custom Search

True Random Number Service

Do you own an iOS or Android device? [Check out our app!](#)

Note: Our much improved new API is currently in public beta - learn more on api.random.org

HTTP Interface Description

RANDOM.ORG is a true random number service that generates randomness via atmospheric noise. This page explains how to interface to the service via the Hyper-Text Transfer Protocol (HTTP). There is also the [HTTP Client Archive](#), which contains clients that other people have written.



Important note!

If you access RANDOM.ORG via an automated client, please make sure you observe the [Guidelines for Automated Clients](#) or your computer may be banned.

If you are writing a general-purpose client, please make sure it is easy for your users to run it in accordance with the guidelines.

This page contains documentation for the [Integer Generator](#), the [Sequence Generator](#), the [String Generator](#) and the [Quota Checker](#), which allows you to examine your current bit allowance.

All the interfaces on this page return HTTP status code 503 (Service Unavailable) in the case of errors and code 200 (OK) when successful. Not all languages allow you to access the HTTP status codes in a straightforward manner. A reasonable workaround is to look for the string "Error:" (don't forget the colon) as the first line of the response. This will work for all the generators on this page, including the [String Generator](#) (which could by chance produce the string "Error" in a successful response, but which cannot produce the colon character).

Please note that the old CGI scripts (randbyte, randnum, etc.) are no longer supported and you should use the ones described below instead. In particular, the old scripts do not return the 503 status code in case of errors (they return the 200 response code in all cases), so please use the new ones instead.

Números aleatórios

```
import java.math.*;

class Main {

    public static void main(String[] args) {

        int i, j, n=30;
        int Matriz[][] = new int[n][n];

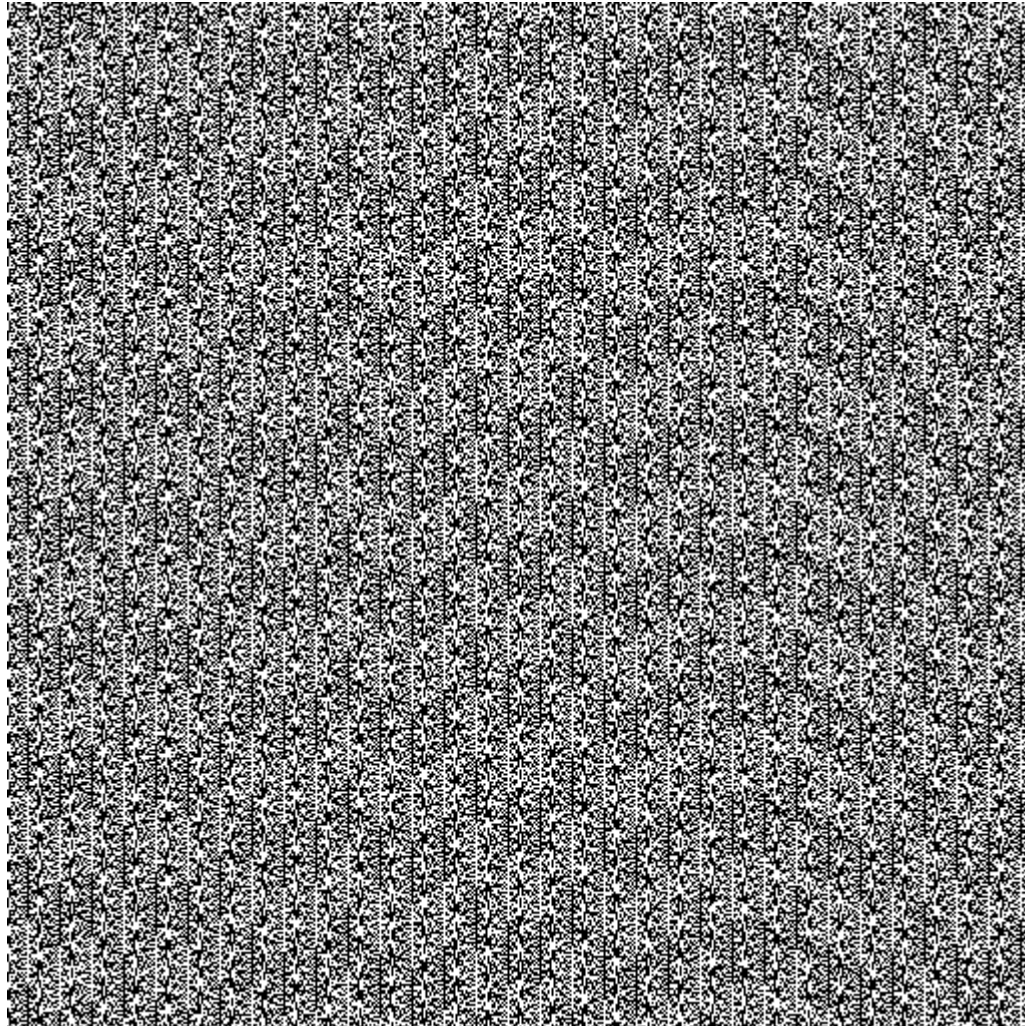
        for (i=0; i<n; i=i+1) {
            for (j=0; j<n; j=j+1) {
                if (Math.random()>0.5)
                    Matriz[i][j] = 1;
            }
        }

        for (i=0; i<n; i=i+1) {
            for (j=0; j<n; j=j+1) {
                System.out.printf("%d ", Matriz[i][j] );
            }
            System.out.printf("\n");
        }
    }
}
```

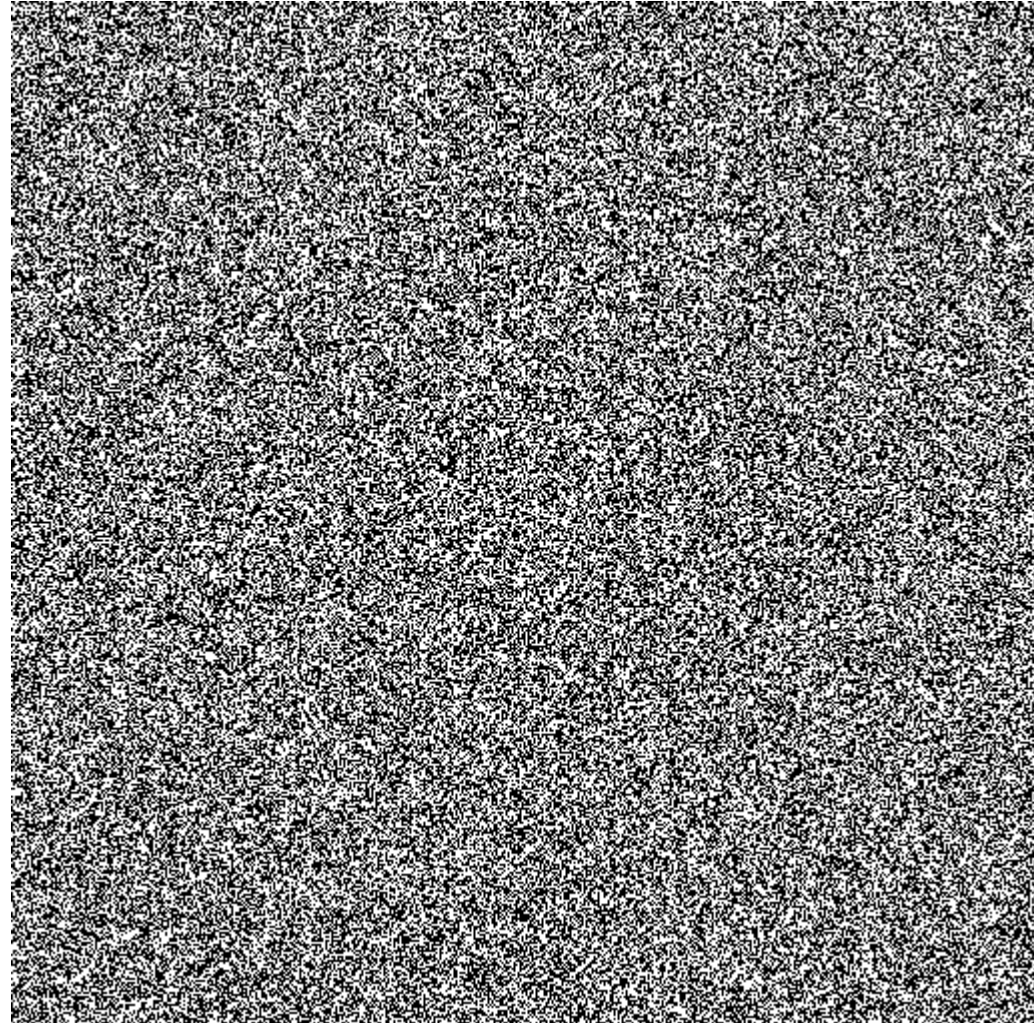
```
0 1 0 1 0 1 1 0 0 0 1 0 0 1 0 0 1 1 0 1 0 1 1 0 0 1 1 1 1 1 0
1 0 1 1 0 0 1 0 1 1 0 1 1 1 0 0 1 0 0 1 1 0 1 1 0 1 0 1 0 0
1 1 0 0 1 0 1 1 0 1 1 0 0 0 0 1 1 1 0 1 0 1 0 1 1 1 0 0 1 1
0 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0
0 0 0 0 0 1 1 1 1 0 1 0 0 0 0 1 1 1 0 0 1 0 1 0 1 1 1 1 1 0
0 1 1 1 1 1 1 1 0 1 1 0 1 0 0 1 1 1 0 1 0 0 0 1 0 0 1 1 0 0
0 1 1 1 0 1 1 1 0 0 1 0 0 1 0 0 1 0 0 1 0 1 0 0 1 0 1 1 0 0 0
1 0 0 0 0 0 1 1 0 1 0 0 1 0 0 1 0 0 0 0 1 1 0 0 0 1 1 0 1 0 0
0 1 0 0 0 0 1 0 1 1 1 1 1 0 0 1 0 1 0 0 1 1 0 1 1 0 0 1 1 0
1 0 0 1 0 0 1 0 0 1 0 0 1 0 1 0 1 0 1 1 1 1 1 1 0 1 0 1 1 1 0
1 1 1 0 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0
0 1 1 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 1 1 1 0 1 0 0 1 1
1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 1 0 0 1 1 1
0 1 0 1 0 1 0 0 0 1 1 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 1 1 0
0 1 1 0 1 1 0 1 0 1 0 1 0 0 1 1 0 0 1 0 1 1 1 1 1 0 1 0 1 1 0
1 1 1 0 0 0 0 1 0 1 1 0 1 0 0 0 0 1 1 0 0 1 1 1 0 1 0 0 1 1
1 0 1 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 0 0 0 1 1 0 0 0 1 0 1 0 1 0
0 0 0 1 1 0 0 0 1 1 0 0 1 1 0 1 1 1 0 0 0 1 0 0 1 1 1 0 1 1
1 0 1 0 0 1 1 1 1 0 0 1 0 1 0 0 0 1 1 1 1 0 1 1 1 0 1 1 1 0
1 1 1 0 1 1 1 1 1 0 1 0 0 1 0 1 0 0 1 1 0 0 1 0 0 1 1 0 1 0
```

Números aleatórios

Pseudo-random



True-random

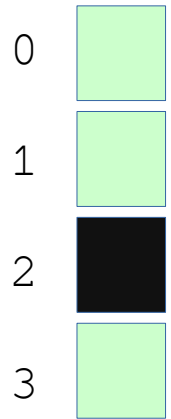




Matrices

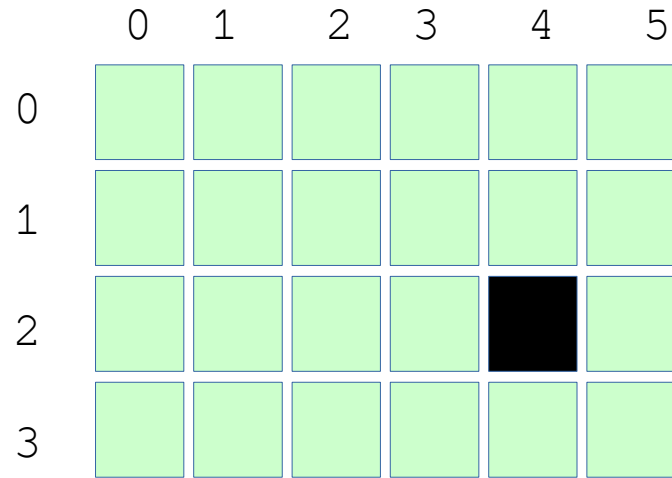
Matrizes

Matriz unidimensional (vetor/Array)



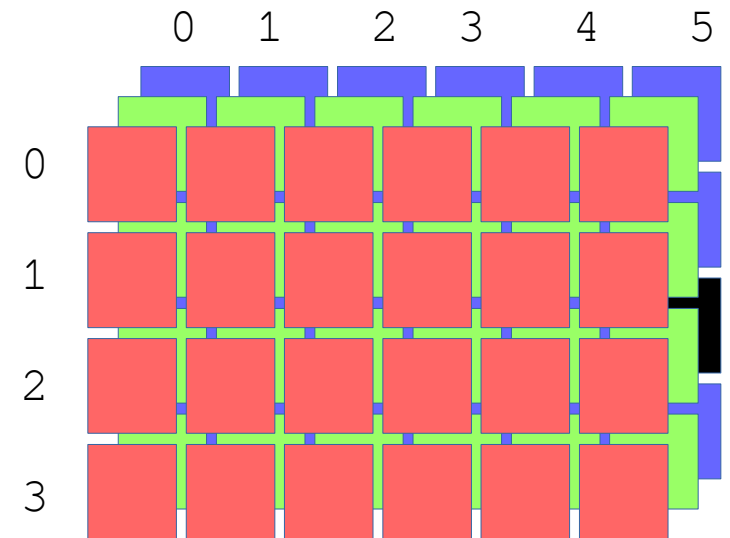
`M[2]`

Matriz bidimensional (2D)



`M[2][4]`

Matriz tridimensional (3D)



`M[2][5][2]`

Exercício 01: Bordas de uma matriz

Vamos desenvolver um método que permita criar uma matriz nula, mas com valores nas bordas iguais a b (dado como entrada).

Assinatura:

```
static int [] [] matrizB( int n, int m, int b )
```

Exemplo:

```
matrizB(4,20,7)
```

```
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7
7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
```

Exercício 01: Bordas de uma matriz

```
static int[][] matrizB( int n, int m, int b ) {
    int i, j;
    int M[][] = new int[n][m];

    for (i=0; i<n; i=i+1) {
        for (j=0; j<m; j=j+1) {
            if (i%(n-1)==0 || j%(m-1)==0) {
                M[i][j] = b;
            }
            else {
                M[i][j] = 0;
            }
        }
    }

    return M;
}
```

Exercício 02: Maior elemento

Vamos desenvolver um método que permita calcular o maior elemento de uma matriz de inteiros.

Assinatura:

```
static int maiorElemento( int [][]M )
```

Exemplo:

7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	777

Resposta 777

Exercício 02: Maior elemento

```
static int maiorElemento( int [][]M ) {
    int i, j;
    int n = M.length;
    int m = M[0].length;
    int maior = M[0][0];

    for (i=0; i<n; i=i+1) {
        for (j=0; j<m; j=j+1) {
            if (M[i][j]>maior) {
                maior = M[i][j];
            }
        }
    }

    return maior;
}
```

Exercício 03: Multiplicação

Vamos desenvolver um método que permita multiplicar cada elemento de uma matriz por um inteiro x .

Assinatura:

```
static int [][] vezes ( int M[] [], int x)
```

Exemplo:

M =

11	11	11
11	11	11

x = 5

Resposta =

55	55	55
55	55	55

Exercício 03: Multiplicação

```
static int [][] vezes ( int M[][], int x) {  
    int i, j;  
    int n = M.length;  
    int m = M[0].length;  
  
    for (i=0; i<n; i=i+1) {  
        for (j=0; j<m; j=j+1) {  
            M[i][j] = M[i][j]*x;  
        }  
    }  
  
    return M;  
}
```

Exercício 04:

Considere M uma matriz de números inteiros.

Qual o valor de K?

```
M = vezes(M, -1);  
k = maiorElemento(M)*(-1);
```

Exercício 04:

Considere M uma matriz de números inteiros.

Qual o valor de K?

```
M = vezes(M, -1);  
k = maiorElemento(M)*(-1);
```

k = menor valor da matriz M.

Exercício 05: Maior elemento – matriz 3D

Vamos desenvolver um método que permita calcular o maior elemento de uma matriz de inteiros.

Assinatura:

```
static int maiorElemento( int M[] [] [] )
```

```
0 1 1 0 0 0 1 0 0 1  
1 1 0 1 0 1 0 0 1 0  
0 1 0 1 1 0 1 0 1 1 0 0  
0 0 0 1 0 0 0 0 1 1 1 0  
1 0 0 1 1 1 1 0 0 0 0 1 0 1  
0 0 1 0 1 0 1 1 1 0 0 0 1 0  
1 1 0 1 0 0 1 0 1 1 0 0 1 1 0  
1 1 0 0 0 1 1 0 1 1 0 0 0 1 0  
0 0 0 1 0 0 1 0 1 1 1 0 0 1 0  
0 1 0 1 0 1 1 1 0 1 0 0 1 1 0 1  
1 0 1 0 1 1 0 0 1 1 1 1 0 1 0 1  
1 1 1 0 0 0 1 1 1 0 0 1 1 1 0 1  
0 0 0 0 0 0 0 1 0 0 0 1 1 0 0  
1 0 1 1 1 1 0 1 1 1 1 0 1 0  
1 0 1 1 0 0 0 0 1 0 0 0 0 1 0  
1 1 1 1 1 0 1 1 1 0 0 1  
1 1 0 1 1 0 0 0 0 1  
0 0 1 0 1 0 1 1 0 0
```

Exercício 05: Maior elemento – matriz 3D

```
static int maiorElemento ( int M[][][] ) {
    int i, j, k;
    int n = M.length;
    int m = M[0].length;
    int p = M[0][0].length;

    int maior = M[0][0][0];

    for (i=0; i<n; i=i+1) {
        for (j=0; j<m; j=j+1) {
            for (k=0; k<p; k=k+1) {
                if (maior<M[i][j][k]) {
                    maior = M[i][j][k];
                }
            }
        }
    }

    return maior;
}
```

Exercício 05: Maior elemento – matriz 3D

```
static int maiorElemento ( int M[][][] ) {
    int i, j, k;
    int n = M.length;
    int m = M[0].length;
    int p = M[0][0].length;

    int maior = M[0][0][0];

    for (i=0; i<n; i=i+1) {
        for (j=0; j<m; j=j+1) {
            for (k=0; k<p; k=k+1) {
                if (maior<M[i][j][k]) {
                    maior = M[i][j][k];
                }
            }
        }
    }

    return maior;
}
```

```
public static void main(String[] args) {
    int i, j, k;
    int Matriz[][][] = new int[10][20][30];
    Matriz[9][11][27] = 999;

    System.out.println( maiorElemento(Matriz) );
}
```