

# Matrici

## (Array 2D, 3D, ...)

# Array bidimensionali (matrici)

- Gli array bidimensionali, o matrici, si possono vedere come “tabelle”. Ad es. voglio:
  - Memorizzare tutte le temperature dell’anno corrente (supponendo non sia bisestile)
  - Accedere poi alla temperatura del 20 ottobre
- Soluzione con array *unidimensionale*:
  - Definire un array **a** di 365 posizioni
  - Contare:  $(31+28+31+30+31+30+31+31+30+20) - 1 = 292$
- Soluzione con array *bidimensionale*:
  - Definire una tabella (matrice) 12x31, poi accedo con  $t[10-1][20-1] = t[9][19]$

# Array bidimensionali (matrici)

	0	1	...	29	30
0					
1					
...					
10					
11					

0	
1	
2	
...	
...	
...	
...	
...	
...	
...	
...	
...	
364	

- Per comodità si potrebbe usare una tabella 13x32 e *ignorare* gli elementi di indice 0
  - Spreco un po' di spazio in memoria, ma codice più leggibile

# Array bidimensionali (matrici)

	0	1	...	30	31
0					
1					
...					
11					
12					

- In questo caso, gli elementi  $a[0]$  e  $t[i][j]$  dove  $i=0$  oppure  $j=0$  non vengono mai utilizzati (“spreco” di memoria)
- Però, l’accesso è più naturale: la temperatura del 20 ottobre viene memorizzata nella cella  $t[10][20]$

0	
1	
2	
...	
...	
...	
...	
...	
...	
...	
365	

# Sintassi Java

- Si usano le “doppie parentesi quadre” per la dichiarazione e l’accesso agli elementi della matrice
  - `double[][] t = new double[13][32]`
  - Il primo numero (13 in questo caso) è il numero di righe, il secondo (32) il numero di colonne
- Doppie quadre → doppio indice
  - `t[10][20]`
  - Il primo è indice di riga, il secondo di colonna
  - **Attenzione:** gli indici partono sempre da **0**

# Matrice

```
// Two rows and three columns.  
int[][] a = new int[2][3];
```

a[0][0]	a[0][1]	a[0][2]
a[1][0]	a[1][1]	a[1][2]

```
final int ROWS = 2;  
final int COLS = 3;  
int[][] board = new int[ROWS][COLS];
```

# Initial Values

```
// Crea e inizializza una matrice 3 x 3  
int[][] tab = {{1,0,0},{0,1,0},{1,2,1}};
```

1	0	0
0	1	0
1	2	1

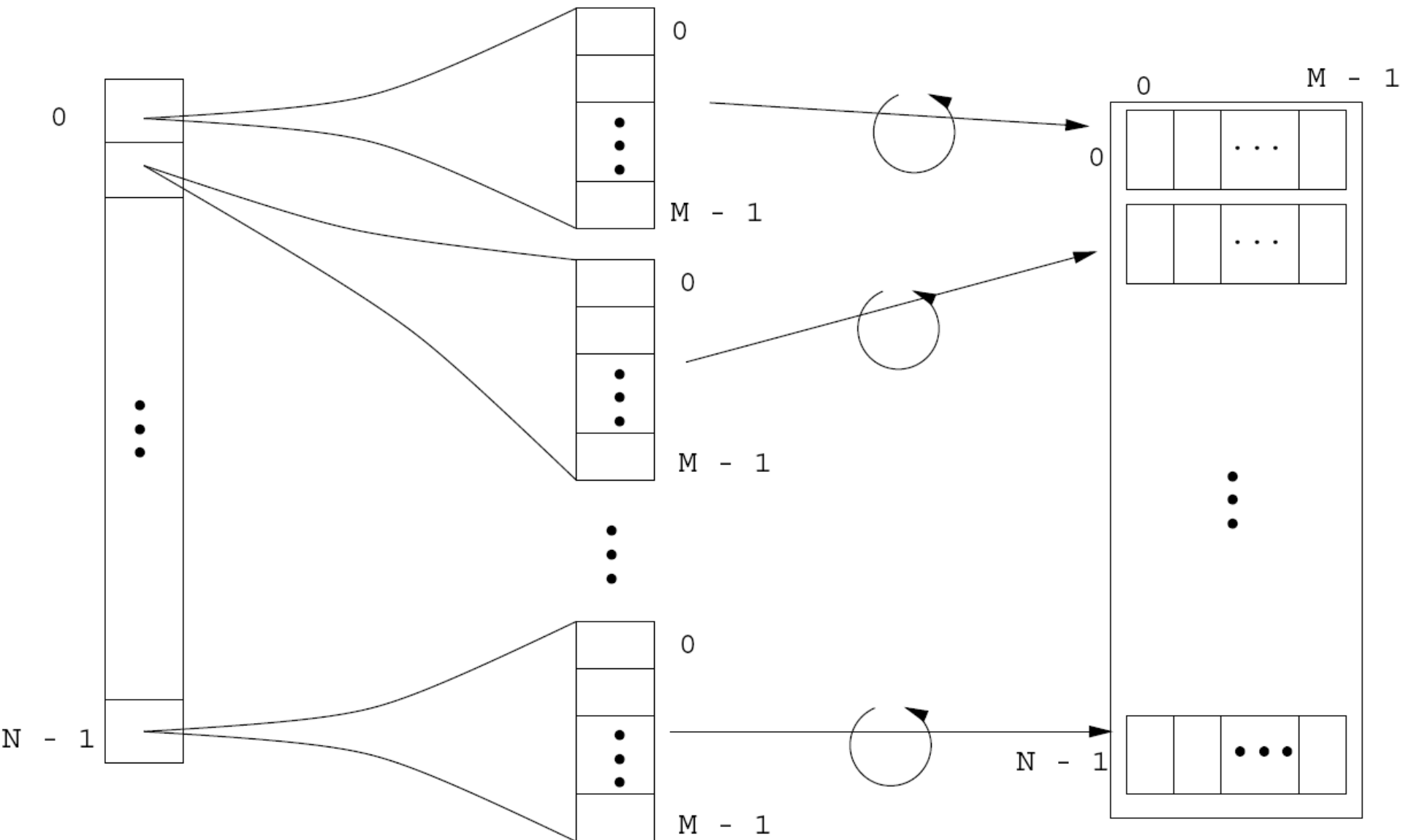
```
// E se invece facessi questo?  
int[][] tab = {{1,0,0,2},{0,1,0},{1,2,1}};
```

# Array di array

- Array bidimensionale =
  - Array i cui elementi sono a loro volta degli array
  - “*Array di array*”
- Per *comodità* lo si rappresenta concettualmente come una matrice/tabella
  - In memoria, non esiste alcuna tabella



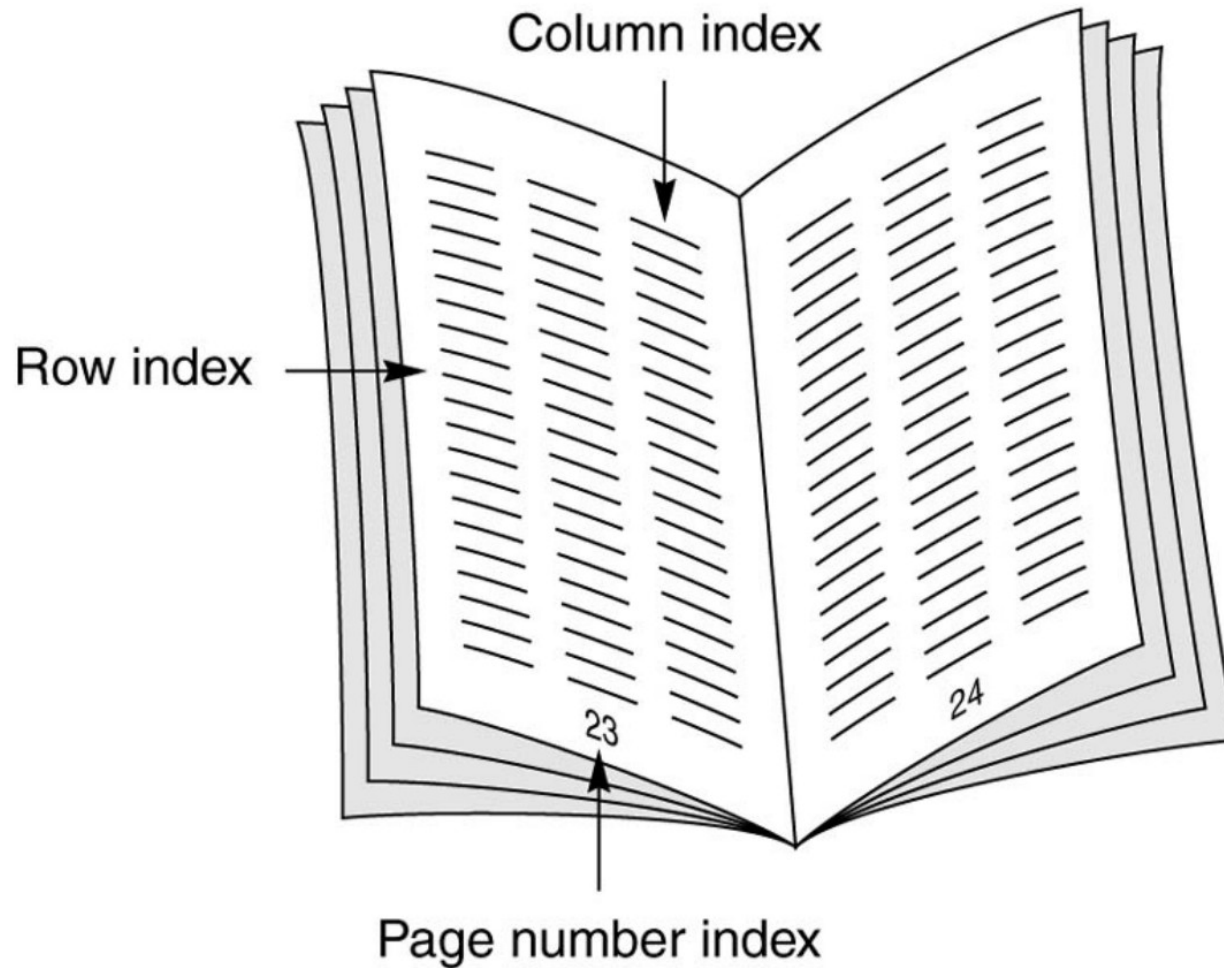
# Array bidimensionali = array di array



# Array multidimensionali

- Array di array di array di ...
- Si usa una coppia di `[]` per ogni dimensione
- Es. array tridimensionali (“cubi”):
  - `double[][][] a;`
  - `a = new double[10][100][1000];`
  - `a[0][0][0] ... a[9][99][999]`
- Utilità
  - Es. registrare la temperatura in diverse ore del giorno
  - `double[][][] t;`  
`t = new double[12][31][24];`  
`// t[mese][giorno][ora]`

# 3d array



# Lavorare su matrici

- Spesso (ma non sempre!) si usano cicli for annidati per scorrere la matrice
  - `.length` di una *matrice* **m** dà il **numero di righe**
  - `.length` su un *elemento* **m[i]** della *matrice* dà il **numero di colonne**
  - Perché?
- Es.: assegnare 1 a tutti gli elem. della matrice

```
for (int i = 0; i < m.length; i++)  
    for (int j = 0; j < m[i].length; j++)  
        m[i][j] = 1;
```

# Numero di Righe e Colonne di una Matrice

- **`m.length`** denota il **numero di righe** della matrice **`m`**
- **`m[i].length`** denota il **numero di colonne** della **riga `i`** della matrice **`m`**

```
double[][] v;  
v = new double[15][20];  
System.out.println(v.length);           // stampa 15  
System.out.println(v[0].length);        // stampa 20
```

# Esempi con matrici

- Esempio: azzerare la riga  $k$ -esima di una matrice bidimensionale  $m$

k	0	0	0	0	0

```
for (int i = 0; i < m.length; i++)  
    for (int j = 0; j < m[i].length; j++)  
        if (i == k)  
            m[i][j] = 0;  
  
for (int j = 0; j < m[k].length; j++)  
    m[k][j] = 0;
```

# Esempi con matrici

- Azzerare la colonna k-esima

			k		
			0		
			0		
			0		
			0		

```
for (int i = 0; i < m.length; i++)  
    for (int j = 0; j < m[i].length; j++)  
        if (j == k)  
            m[i][j] = 0;  
  
for (int i = 0; i < m.length; i++)  
    m[i][k] = 0;
```

# Esempi con matrici

- Azzerare la diagonale principale
  - Esercizio: azzerare la diagonale *secondaria*

	0				
		0			
			0		
				0	
					0

```
for (int i = 0; i < m.length; i++)  
    for (int j = 0; j < m[i].length; j++)  
        if (i == j)  
            m[i][j] = 0;  
  
for (int i = 0; i < m.length; i++)  
    m[i][i] = 0;
```



# Esercizi

- Data una matrice (bilanciata) di interi:
  - 1) Calcolare la somma degli elementi nella riga  $k$ -esima
  - 2) Calcolare il numero di righe che contengono almeno un valore 0
  - 3) Calcolare il numero di colonne che contengono almeno due valori pari
  - 4) Calcolare il numero di righe per cui la somma degli elementi è maggiore di 100