

Corso di Laurea Ingegneria Informatica

Fondamenti di Informatica 1

Dispensa 03

Array di array

A. Miola

Febbraio 2008

Contenuti

□ **Array bidimensionali**

- **creazione e uso di array bidimensionali**

□ **Array di array**

□ **Gestione di array di array**

- **visualizzazione degli elementi di un array di array**
- **somma degli elementi di un array di array**
- **uguaglianza tra array di array**

□ **Esercizi**

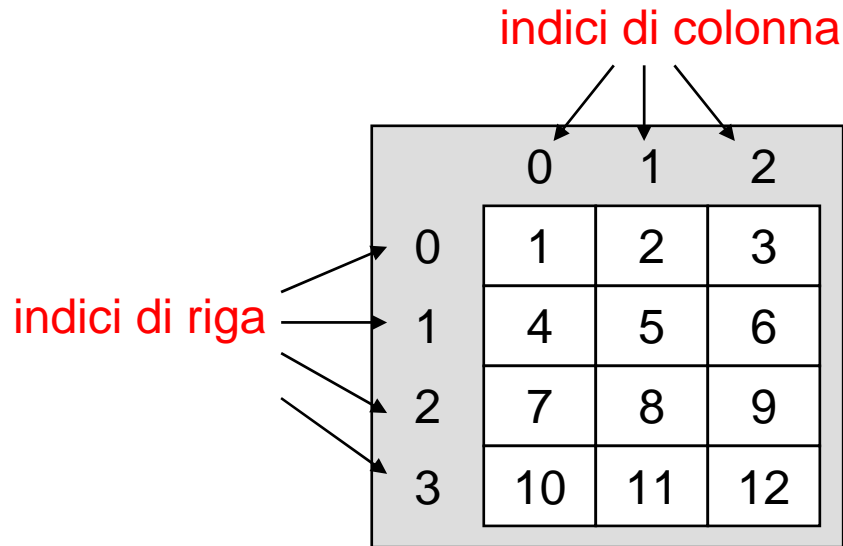
Array di array

- ❑ **Per molte applicazioni, le informazioni di interesse devono essere rappresentate da collezioni di variabili omogenee organizzate secondo una struttura a più dimensioni**
 - **ad esempio, matrici numeriche, disposizione dei pezzi su una scacchiera, ...**
- ❑ **Nei linguaggi di programmazione, uno dei supporti per la gestione di dati a due o più dimensioni è fornito dagli array di array**
 - **un array di array è un array i cui elementi sono a loro volta di tipo array**

Array bidimensionali

- Un **array bidimensionale** è una collezione finita di variabili di uno stesso tipo, posto in **corrispondenza biunivoca con il prodotto cartesiano di una coppia di insiemi finiti**
 - ciascun **elemento** di un array bidimensionale è **individuato** dal riferimento all'array e da una **coppia di indici**
 - il **primo** indice è relativo alla **riga** in cui si trova l'elemento
 - il **secondo** indice è relativo alla **colonna** in cui si trova l'elemento
- In Java, un **array bidimensionale** è una collezione finita di variabili di uno stesso tipo, posto in corrispondenza biunivoca con il prodotto cartesiano di una coppia di intervalli iniziali e finiti dei numeri naturali
 - l'**accesso** agli elementi di un array bidimensionale avviene **usando** come indici una **coppia di numeri naturali**

Array bidimensionali



- **Gli array bidimensionali (e più in generale gli array di array) consentono di gestire un gruppo di variabili omogenee che possono essere organizzati in modo naturale mediante una struttura a griglia bidimensionale**

Creazione e uso di array bidimensionali

- Per usare un array bidimensionale i cui elementi sono di un tipo **T**, bisogna dichiarare una variabile di tipo **T[][]**

```
int[][] numeri; // un array bidimensionale di numeri
```

- Per creare un array bidimensionale composto da **R** righe e **C** colonne di elementi di tipo **T** bisogna usare l'espressione **new T[R][C]**

```
/* creazione l'array bidimensionale numeri */  
final int RIGHE = 4;  
final int COLONNE = 3;  
numeri = new int[RIGHE][COLONNE];
```

Creazione e uso di array bidimensionali

□ L'accesso a un elemento di un array bidimensionale

- il riferimento all'array
- una coppia di espressioni intere – **indice di riga** e **indice di colonna** – tra parentesi quadre [e]
- esempio

```
numeri[0][0] = 1;
```

Scansione per righe

□ Di solito bisogna elaborare tutti gli elementi di un array bidimensionale

▪ scansione per righe

- prima tutti gli elementi della prima riga, poi quelli della seconda riga, e così via
- due variabili indice e due istruzioni ripetitive annidate

```
int i, j; // indici per la scansione di numeri
```

```
/* scansione per righe degli elementi di numeri */
```

```
for (i=0; i<RIGHE; i++)
```

```
    for (j=0; j<COLONNE; j++)
```

```
        ... elabora numeri[i][j] ...
```


Calcola gli elementi di numeri

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9
3	10	11	12

```
int i, j; // indici per la scansione di numeri
/* assegna un valore agli elementi di numeri */
for (i=0; i<RIGHE; i++)
    for (j=0; j<COLONNE; j++)
        numeri[i][j] = i*COLONNE+j+1;
```

Visualizzazione degli elementi di numeri

```
int i, j;    // indici per la scansione di numeri
/* visualizza gli elementi di numeri */
for (i=0; i<RIGHE; i++) {
    /* visualizza la riga i-esima di numeri */
    for (j=0; j<COLONNE; j++)
        System.out.print(neri[i][j]+ " ");
    System.out.println();
}
```

viene visualizzato

1 2 3

4 5 6

7 8 9

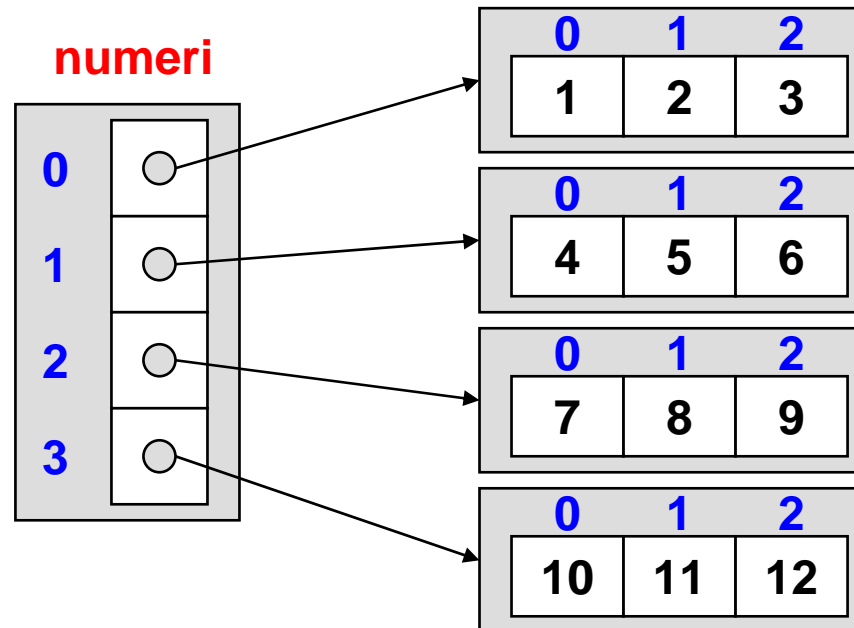
10 11 12

Esercizi

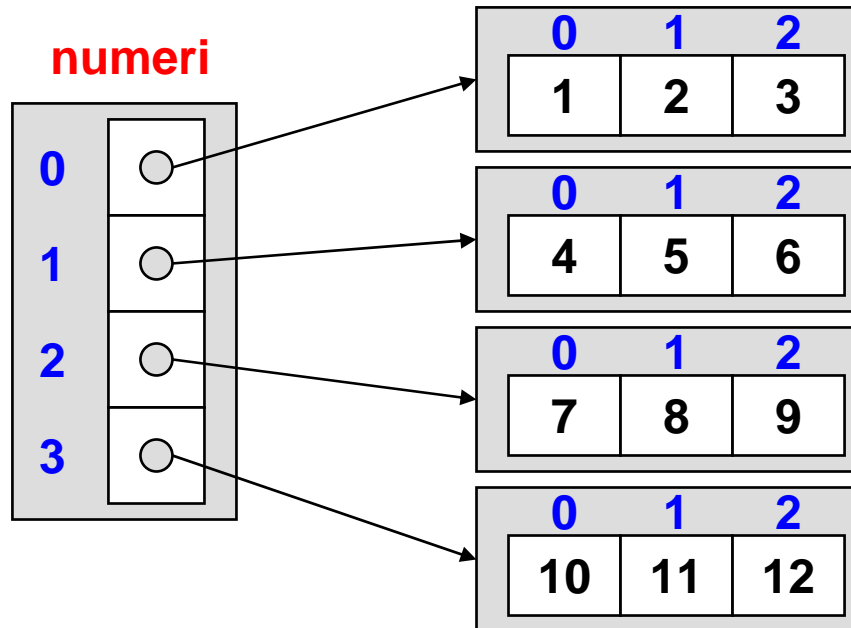
- ❑ Scrivere un metodo `int[][] identita(int n)` che calcola un array bidimensionale che rappresenta la matrice identità di ordine `n`
 - `n` è un numero naturale
- ❑ Scrivere un metodo `boolean diagonale(int[][] m)` che verifica se un array bidimensionale `m` sia una matrice diagonale

Array di array

- Un **array di array** è un array i cui elementi sono a loro volta di tipo array
 - **gli array bidimensionali sono un caso particolare degli array di array**

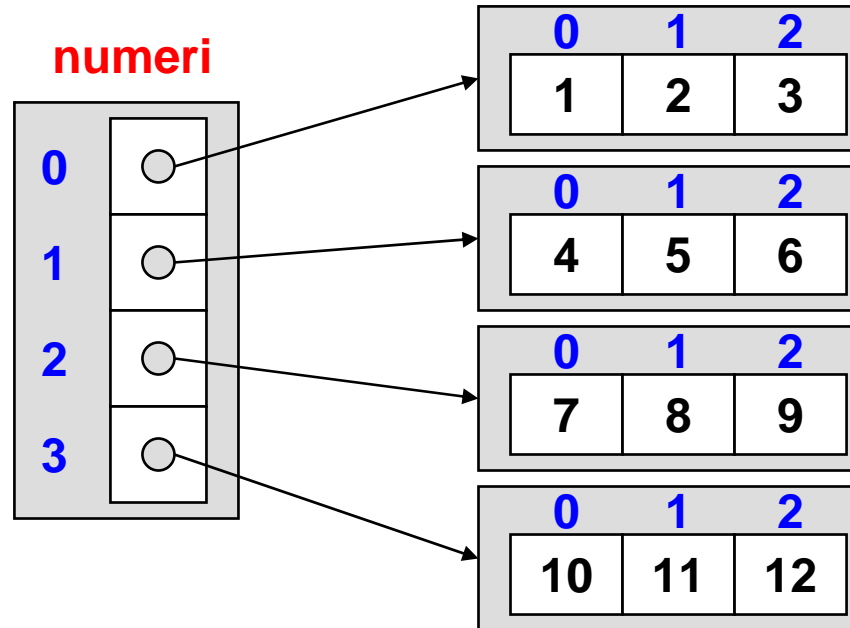


Gli array bidimensionali sono array di array



- **numeri.length** è la lunghezza dell'array **numeri**
 - in un array bidimensionale **M**, l'espressione **M.length** denota il **numero di righe** di **M**
- **numeri[0]** è una espressione valida — che riferenzia la prima riga di **numeri**, un array di interi

Gli array bidimensionali sono array di array



- **numeri[0].length** è una espressione valida — è la lunghezza della prima riga di **numeri**, che vale 3
 - in un array bidimensionale **M**, l'espressione **M[0].length** denota il **numero di colonne** di **M**

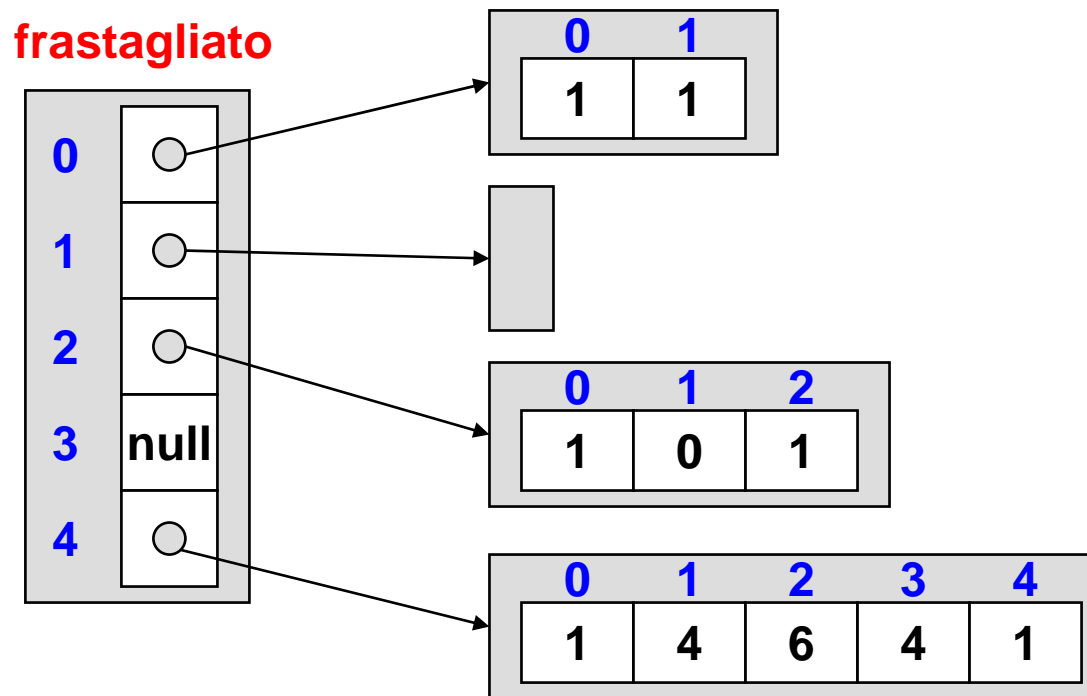
Tipo delle espressioni array

□ Sia **M** un array di array di interi

- **M** denota un array di array di interi
 - **M** è una espressione di tipo **int[][]**
 - la notazione **int[][]** indica un array di elementi di tipo **int[]**
- **M.length** denota la lunghezza di **M**, ovvero il numero di array che compone l'array di array **M**
- **M[0]** denota un array di interi
 - ovvero, **M[0]** è una espressione di tipo **int[]**
- **M[0].length** denota la lunghezza dell'array di interi **M[0]**
 - è possibile accedere alla variabile **length** di un array, che vale la lunghezza dell'array
 - in modo simile, **M[1].length** denota la lunghezza dell'array di interi **M[1]**
- **M[0][1]** denota un intero
 - ovvero, **M[0][1]** è una espressione di tipo **int**

Array di array frastagliati

- ❑ Un array di array che non è un array bidimensionale



Letterali array di array

- Un letterale array di array è la denotazione di un array di array costante – **ad esempio**

```
int[][] numeri;  
numeri = new int[][] { { 1, 2, 3 },  
                        { 4, 5, 6 },  
                        { 7, 8, 9 } ,  
                        { 10, 11, 12 } };
```

```
int[][] frastagliato;  
frastagliato = new int[][] { { 1, 1 }, { },  
                             { 1, 0, 1 }, null,  
                             { 1, 4, 6, 4, 1 } };
```

Visualizzazione di un array di array

□ Visualizzazione per righe

```
/* Visualizza gli elementi dell'array di array a. */
public static void visualizza(int[][] a) {
    // pre: a!=null && ogni riga di a è non nulla
    int i, j;    // indici per la scansione di a

    /* visualizza gli elementi di a */
    for (i=0; i<a.length; i++) {
        /* visualizza la riga i-esima di a */
        for (j=0; j<a[i].length; j++)
            System.out.print(a[i][j] + " ");
        System.out.println(); }
}
```

osservare l'uso della variabile **length** nelle condizioni delle istruzioni ripetitive

Esercizio

- **Scrivere un metodo che, ricevendo come parametro un array di array A, verifica se A è bidimensionale**

Somma degli elementi di un array di array

```
/* Calcola la somma degli elementi dell'array di array a. */
public static int somma(int[][] a) {
    // pre: a!=null && ogni riga di a è non nulla
    int i, j;    // indici per la scansione di a
    int s;      // somma degli elementi di a

    /* calcola la somma degli elementi di a */
    s = 0;
    for (i=0; i<a.length; i++)
        /* somma a s gli elementi della riga i-esima di a */
            for (j=0; j<a[i].length; j++)
                s += a[i][j];
    return s;
}
```

Somma degli elementi di un array di array

- La somma degli elementi di un array di array può essere anche calcolata come la somma delle somme degli elementi delle righe di un array di array

```
/* Calcola la somma degli elementi dell'array di array a. */
public static int somma(int[][] a) {
    // pre: a!=null && ogni riga di a è non nulla
    int i;    // indice per la scansione delle righe di a
    int s;    // somma degli elementi di a
    /* calcola la somma degli elementi di a come somma
     * delle somme degli elementi delle righe di a */
    s = 0;
    for (i=0; i<a.length; i++)
        s += somma(a[i]);
    /* usa int somma(int[] a) come metodo di supporto */
    return s;
}
```

Esercizio

- **Scrivere un metodo che, ricevendo come parametro un array di array A in cui ciascuna riga è non vuota, calcola e restituisce un array MAX tale che**
 - **la lunghezza di MAX è uguale al numero di righe di A**
 - **ogni elemento di MAX è uguale al massimo della riga corrispondente di A**

Uguaglianza tra array di array

```
/* Verifica se gli array di array a e b sono uguali. */
public static boolean uguali(int[][] a, int[][] b) {
    // pre: a!=null && b!=null &&
    //       gli elementi di a e b sono non nulli
    int i;           // indice per la scansione di a e b
    boolean uguali; // a e b sono uguali
    /* verifica se a e b sono uguali */
    if (a.length!=b.length) // sono sicuramente diversi
        uguali = false;
    else { // sono uguali se lo sono riga per riga
        uguali = true;
    }
    /* usa boolean uguali(int[] a, int[] b) come metodo di supporto */
    for (i=0; uguali && i<a.length; i++)
        if (!uguali(a[i],b[i]))
            uguali = false;
    }
    return uguali;
}
```

Esercizi

- ❑ **Scrivere un metodo che, dato un array di array A, verifica se A contiene almeno una coppia di righe uguali**

- ❑ **Scrivere un metodo che, dato un array di array A, verifica se A contiene almeno una coppia di elementi tra loro uguali**

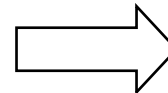
Esercizio

- Siano A e B array bidimensionali di interi, di dimensione rispettivamente $N \times M$ e $M \times H$
- Scrivere un metodo che, data una coppia di array bidimensionali A e B tale che il numero di colonne di A è uguale al numero di righe di B , calcola e restituisce la matrice prodotto righe per colonne di A per B

1	2
0	1
0	-1

x

2	0	1	-1
2	1	0	-1



6	2	1	-3
2	1	0	-1
-2	-1	0	1

Esercizio

- ❑ Scrivere un metodo che, dato un array di array di interi A, verifica se tutti gli elementi di A che valgono 1 formano un rettangolo

0	0	0	0	0	4	0	0
0	0	0	1	1	1	1	0
0	7	0	1	1	1	1	0
0	0	0	1	1	1	1	0
0	9	21	0	0	0	6	0

Esercizio

- ❑ Scrivere un metodo che, dato un numero positivo N , crea e restituisce una matrice quadrata $N \times N$ di interi che memorizza i numeri da 1 a N^2 disposti “a spirale”
 - ad esempio, le matrici a spirale di ordine 3 e 4

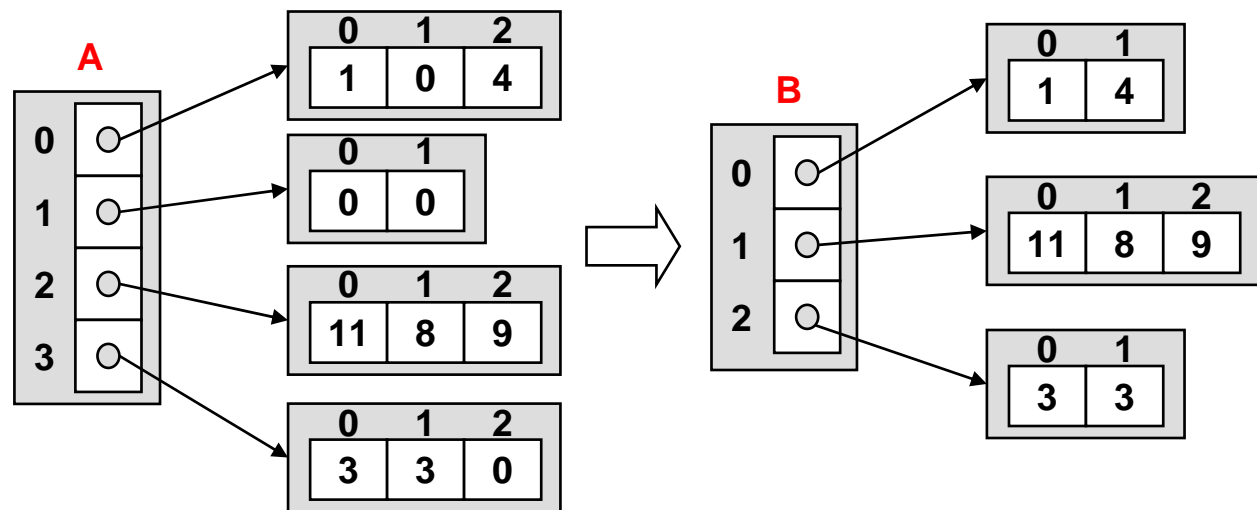
1	2	3
8	9	4
7	6	5

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

Esercizio

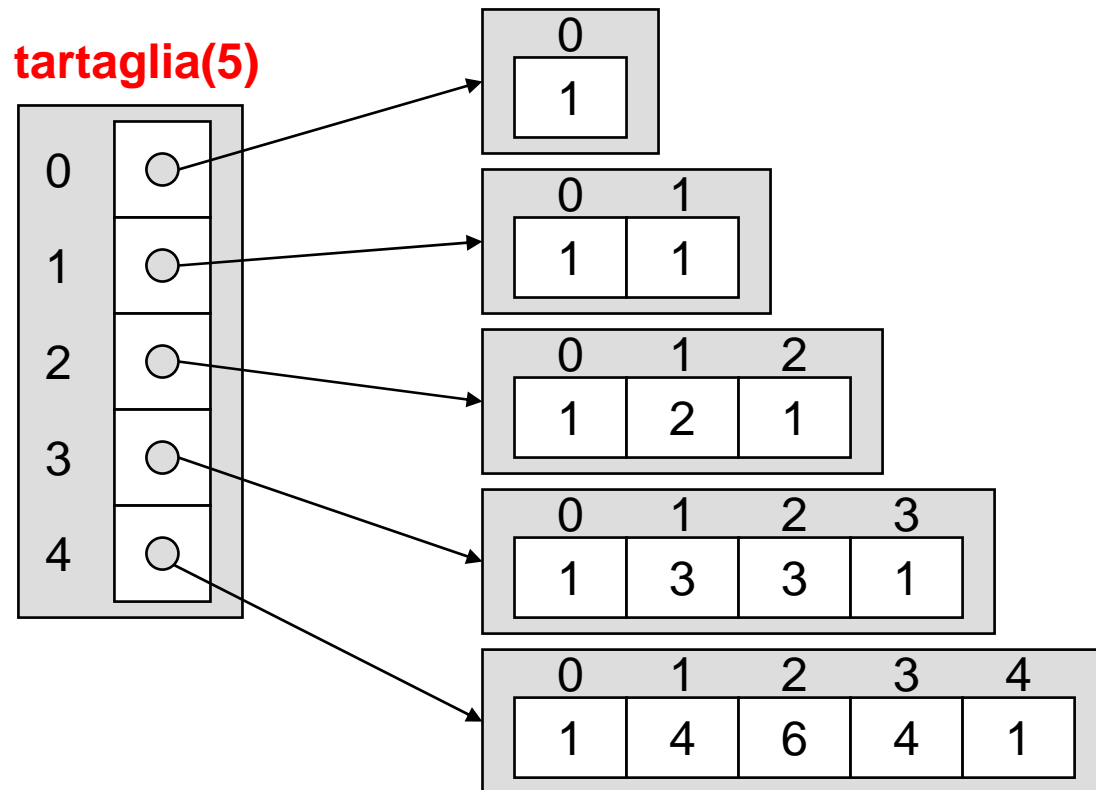
□ Scrivere un metodo che, dato un array di array di interi A, crea e restituisce un nuovo array di array di interi B ottenuto da A

- eliminando tutti gli elementi che valgono 0
- tutte le righe vuote



Esercizio

- ❑ Scrivere un metodo `int[][] tartaglia(int n)`, con `n` naturale, che calcola le prime `n` righe del triangolo di Tartaglia



Riferimenti al libro di testo

- ❑ Per lo studio di questi argomenti si fa riferimento al libro di testo, e in particolare al **capitolo 20**