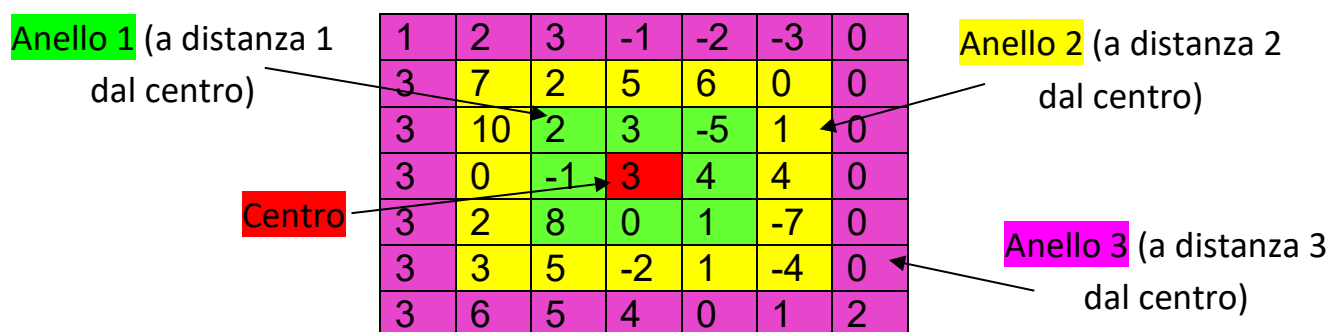


## Esercizio Java n. 1: Anelli Bilanciati

Esercizio estratto e adattato da Compito VI Appello – 09/02/2023

Sia  $T$  una matrice quadrata di interi di dimensione  $m \times m$ , con  $m \geq 3$  e  $m$  dispari. Sia  $k$  un intero, con  $1 \leq k \leq m/2$ .

Informalmente, si definisce anello  $k$ -mo della matrice  $T$  come l'insieme di celle di  $T$  che “distano”  $k$  celle dal centro della matrice. Ad esempio, in una matrice  $T$  di dimensione  $7 \times 7$  si possono definire tre anelli a distanza 1, 2 e 3 dal centro, come nella seguente figura (esempio 1):



L'anello  $k$  della matrice  $T$  si dice bilanciato se la somma di ciascuna coppia di elementi “opposti” dell'anello  $k$  è uguale all'elemento centrale della matrice.

Ad esempio, l'anello 1 della matrice dell'esempio precedente è bilanciato poiché  $2+1=3+0=-5+8=4-1=3$  (centro della matrice):

1	2	3	-1	-2	-3	0
3	7	2	5	6	0	0
3	10	2	3	-5	1	0
3	0	-1	3	4	4	0
3	2	8	0	1	-7	0
3	3	5	-2	1	-4	0
3	6	5	4	0	1	2

Nello stesso esempio:

- l'anello 3 della matrice è bilanciato (la somma di ogni coppia di elementi opposti è uguale a 3);
- l'anello 2 della matrice NON è bilanciato, poiché esiste almeno una coppia di elementi opposti la cui somma è diversa da 3 (in figura sono

stati evidenziate in giallo le due coppie di elementi opposti dell'anello 2 che non hanno somma uguale a 3, ovvero  $0+4 \neq 3$  e  $6+5 \neq 3$ ).

**[DIFFICOLTA' STANDARD]:** Scrivere un metodo Java-- chiamato `anelloBilanciato` che, dati in input una matrice quadrata  $T$  di interi di dimensione  $m \times m$  (con  $m \geq 3$  e  $m$  dispari), ed un intero  $k$  (con  $1 \leq k \leq m/2$ ), restituisca `true` se l'anello  $k$  della matrice  $T$  è bilanciato, `false` altrimenti.

**[DIFFICOLTA' RIDOTTA]:** Si può svolgere l'esercizio precedente assumendo che  $k$  sia sempre minore o uguale a 2, ovvero che  $1 \leq k \leq 2$ .

NOTA BENE:

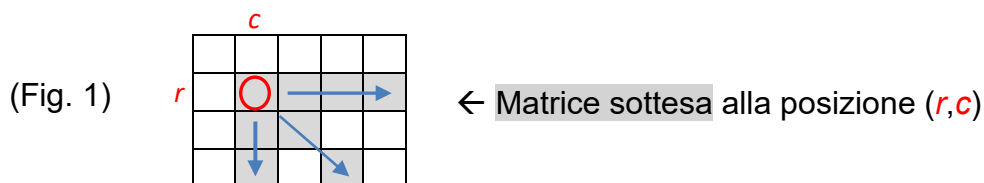
- I Junit Test da superare per la consegna con difficoltà ridotta sono quelli della classe **AnelloBilanciatoTest\_Ridotta** (quelli della classe `AnelloBilanciatoTest_Standard` falliranno).
- I Junit Test da superare per la consegna con difficoltà standard sono quelli della classe **AnelloBilanciatoTest\_Standard**.
- Nello svolgere l'esercizio NON devono essere utilizzati i metodi `clone`, o `arraycopy`, o metodi della classe `Arrays`. L'utilizzo di tali metodi renderà l'esercizio automaticamente insufficiente.

## Esercizio Java n. 2: azzerare non unici

Esercizio estratto e adattato da compito V Appello del 23-01-2023

Sia  $M$  una matrice di interi di  $m$  righe ed  $n$  colonne, con  $m > 0$  e  $n > 0$ , e siano  $r$  e  $c$ , con  $0 \leq r < m$  e  $0 \leq c < n$ , due numeri interi che rappresentano, rispettivamente, un indice riga e un indice colonna della matrice. La procedura “azzerare non unici” applicata all’elemento della matrice  $M$  di posizione  $(r, c)$  è una procedura che modifica i valori della matrice nel seguente modo:

1. Conta il numero di occorrenze del valore  $M(r, c)$  nella sua “matrice sottesca”, ovvero nella seguente porzione di matrice (si veda Fig. 1):
  - nella colonna  $c$  partendo dalla riga  $r$ , spostandosi verso il basso (verso indici di riga crescenti);
  - nella riga  $r$  partendo dalla colonna  $c$ , spostandosi verso destra (verso indici di colonna crescenti);
  - nella diagonale a partire dall’elemento di riga  $r$  e colonna  $c$ , spostandosi lungo la diagonale di direzione basso-destra (verso indici di riga e colonna crescenti).



2. Se il numero di occorrenze del valore  $M(r, c)$  nella sua matrice sottesca (calcolato al punto 1) è maggiore o uguale a due (quindi  $M(r, c)$  non è unico), allora si pongono uguali a zero tutte le sue occorrenze. Se invece c’è solo una occorrenza del valore  $M(r, c)$ , la matrice non viene modificata.

Ad esempio, sia  $M = \begin{pmatrix} 5 & 0 & 17 & 8 \\ 9 & 10 & 11 & 12 \\ 10 & 9 & 10 & 12 \\ 9 & 17 & 9 & 0 \end{pmatrix}$ , e sia  $(r, c) = (1, 0)$ , (esempio 1)

l’applicazione della procedura “azzerare non unici” all’elemento di posizione  $(1, 0)$  azzererà tutte le occorrenze del valore 9 nella matrice sottesca poiché la matrice sottesca contiene 4 occorrenze del valore 9. La matrice  $M$  modificata sarà quindi la seguente:

$$M' = \begin{pmatrix} 5 & 0 & 17 & 8 \\ 0 & 10 & 11 & 12 \\ 10 & 0 & 10 & 12 \\ 0 & 17 & 0 & 0 \end{pmatrix}$$

Nello stesso esempio 1, l’applicazione della procedura “azzerare non unici” all’elemento di posizione  $(0, 2)$  della matrice  $M$  non modificherà la matrice, in quanto il valore 17 compare una sola volta nella sua matrice sottesca (è unico):

$$M=M'=\begin{pmatrix} 5 & 0 & 17 & 8 \\ 9 & 10 & 11 & 12 \\ 10 & 9 & 10 & 12 \\ 9 & 17 & 9 & 0 \end{pmatrix}$$

### Consegna Minima (OBBLIGATORIA):

Scrivere un metodo Java-- chiamato `azzeraNonUnici` che, data una matrice di interi  $M$  di  $m$  righe ed  $n$  colonne, con  $m>0$  e  $n>0$ , e dati due numeri interi  $r$  e  $c$ , con  $0\leq r<m$  e  $0\leq c<n$ , restituisca la matrice  $M$  modificata applicando la procedura “azzera non unici” all’elemento di posizione  $(r,c)$ .

### Consegna Standard (FACOLTATIVA):

Scrivere un metodo Java-- chiamato `azzeraNonUniciMatrice` che applica la procedura “azzera non unici” in sequenza a tutti gli elementi della matrice  $M$ , visitando la matrice dalla prima riga (riga 0) verso l’ultima (riga  $m-1$ ), scorrendo gli elementi di ciascuna riga da sinistra (colonna 0) verso destra (colonna  $n-1$ ).

Ad esempio, considerando la stessa matrice  $M$  dell’esempio 1 e applicando in sequenza la procedura a tutti i suoi elementi si ottiene la matrice così modificata:

$$M'=\begin{pmatrix} 5 & 0 & 17 & 8 \\ 0 & 0 & 11 & 0 \\ 10 & 0 & 0 & 0 \\ 0 & 17 & 0 & 0 \end{pmatrix}$$

**JUnit Test:** I JUnit Test che devono essere superati sono i seguenti:

- per la consegna minima (obbligatoria): test della classe **AzzeraNonUnici\_Test**. Se non verrà svolta la consegna standard (facoltativa), non considerare i test della classe **AzzeraNonUniciMatrice\_Test** che ovviamente falliranno.
- per la consegna standard (facoltativa): test della classe **AzzeraNonUniciMatrice\_Test** (anche i test della classe **AzzeraNonUnici\_Test** devono essere superati).

### **NOTA BENE:**

Nello svolgere l’esercizio NON devono essere utilizzati i metodi `clone`, o `arraycopy`, o metodi della classe `Arrays`. L’utilizzo di tali metodi renderà l’esercizio automaticamente insufficiente.