

## Esercizio Java n. 1: Compatta Matrice

*Esercizio estratto e adattato da – Compito VI Appello – del 11/02/2021.*

Sia  $T$  una matrice di numeri interi di dimensione  $m \times n$  (con  $m > 0$  e  $n > 0$ ). La matrice può contenere un numero arbitrario di elementi uguali a zero (anche tutti gli elementi della matrice possono essere uguali a zero, o anche nessun elemento).

Siano  $T[r_1, c_1], T[r_2, c_2], \dots, T[r_k, c_k]$  tutti i  $k$  numeri interi della matrice  $T$  diversi da zero (se esistono), che si incontrano visitando la matrice dalla prima riga (di indice 0) verso l'ultima e dalla prima colonna (di indice 0) verso l'ultima.

La “*forma compatta*” della matrice  $T$  corrisponde a una nuova matrice  $C$  di dimensione  $k \times 3$ , in cui ciascuna riga  $j$  contiene i seguenti tre elementi: il valore  $T[r_j, c_j]$ , il corrispondente indice riga ( $r_j$ ) ed il corrispondente indice colonna ( $c_j$ ), per  $j = 1, 2, \dots, k$ , ovvero:

$$C = \begin{pmatrix} T[r_1, c_1] & r_1 & c_1 \\ T[r_2, c_2] & r_2 & c_2 \\ \vdots & \vdots & \vdots \\ T[r_k, c_k] & r_k & c_k \end{pmatrix}.$$

Scrivere un metodo Java-- chiamato `compattaMatrice` che, dato in input una matrice  $T$  di interi di dimensione  $m \times n$  (con  $m > 0$  e  $n > 0$ ) contenente un numero arbitrario di elementi uguali a zero, restituisca la “*forma compatta*”  $C$  della matrice  $T$ . Nota: se tutti gli elementi della matrice  $T$  sono uguali a zero, il metodo deve restituire `null`.

Ad esempio, se  $T = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 5 & 3 & 0 & -2 \\ 0 & 0 & 0 & 0 \\ 0 & -6 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$ , allora  $C = \begin{pmatrix} 5 & 2 & 0 \\ 3 & 2 & 1 \\ -2 & 2 & 3 \\ -6 & 4 & 1 \end{pmatrix}$

Infatti: visitando la matrice dall'alto verso il basso e per ogni riga da sinistra verso destra, si incontra 5 come primo elemento di  $T$  diverso da zero e si trova nella cella di riga 2 e colonna 0; poi si incontra 3, che si trova a riga 2 e colonna 1; poi -2, in riga 2 e colonna 3, ed infine il numero -6 che si trova a riga 4 e colonna 1.

**NOTA BENE:** Nello svolgere l'esercizio non si devono utilizzare metodi o librerie di ordinamento già esistenti in Java. **Eventuali metodi di ordinamento devono essere sviluppati dagli studenti.** L'utilizzo di metodi/librerie di ordinamento già esistenti renderà insufficiente l'esercizio.

## Esercizio Java n. 2: Occorrenze Compresso

*Esercizio estratto da Raccolta 50 esercizi Java--.*

Dato un array  $a$  di  $n$  interi non negativi, l'array delle occorrenze  $o$  è un array di  $m+1$  elementi (dove  $m$  è il valore massimo di  $a$ ) tale che  $o[i]$  è il numero di volte che il valore  $i$  occorre in  $a$ . Ad esempio, dato l'array  $\{1, 0, 3, 4, 1, 4\}$ , l'array  $o$  sarà  $\{1, 2, 0, 1, 2\}$  che ha 5 elementi perché il valore massimo in  $a$  è 4; inoltre, in  $a$ , 0 occorre una volta, 1 occorre 2 volte, 2 non occorre, 3 occorre una volta e 4 occorre 2 volte.

L'array delle occorrenze compresso  $oc$  è la versione compressa di  $o$  cioè senza gli elementi con valore 0: ad esempio, l'array  $oc$  che corrisponde alla versione compressa del precedente array  $o$  è  $\{1, 2, 1, 2\}$ .

Scrivere un metodo Java-- chiamato `occorrenzeCompresso` che dato un array  $a$  restituisce un nuovo array che corrisponde a  $oc$ .