

Esercizio Java n. 1: Scorrimento righe di matrice

Estratto e adattato da Compito I Appello del 29/05/2023.

Sia M una matrice di interi di dimensione $m \times n$, con $m > 0$ e $n > 0$. Uno scorrimento delle righe della matrice M modifica la stessa matrice spostando di 1 posizione verso destra (in modo circolare) tutti gli elementi delle righe di indice pari (0, 2, 4, 6, ...), e di 1 posizione verso sinistra tutti gli elementi delle righe di indice dispari (1, 3, 5, ...). Ad esempio, sia M la matrice:

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

Lo scorrimento delle righe della matrice M modifica la stessa matrice in questo modo:

		7	1	2	3	4	5	6		
		9	10	11	12	13	14	8		
		21	15	16	17	18	19	20		
		23	24	25	26	27	28	22		

Righe di indice dispari: scorrimento di 1 passo verso sinistra

Righe di indice pari: scorrimento di 1 passo verso destra

[CONSEGNA OBBLIGATORIA]: Scrivere un metodo Java-- di nome `scorrimentoRighe` che, dati in input una matrice M di interi di dimensione $m \times n$, con $m > 0$ e $n > 0$, restituisca la stessa matrice modificata da uno scorrimento delle sue righe come precedentemente descritto.

[CONSEGNA EXTRA - FACOLTATIVA]: Sia p un numero intero di tipo `short`, con $p > 0$. Lo scorrimento di p posizioni delle righe della matrice M modifica la stessa matrice applicando consecutivamente p volte lo scorrimento delle righe della matrice come precedentemente descritto. In altre parole, si spostano di p posizioni verso destra tutti gli elementi delle righe di indice pari, e di p posizioni verso sinistra tutti gli elementi delle righe di indice dispari). Scrivere un metodo Java-- di nome `scorrimentoRigheNposizioni` che, dati in input una matrice M di interi di dimensione $m \times n$, con $m > 0$ e $n > 0$, ed un numero intero $p > 0$ di tipo `short`, restituisca la stessa matrice modificata da uno scorrimento di p posizioni delle sue righe come precedentemente descritto.

NOTA BENE:

- Saranno premiate le soluzioni che occuperanno meno memoria dati.
- I Junit Test da superare per la consegna obbligatoria sono quelli della classe **MatriceTraslaRigheTest** (quelli della classe `MatriceTraslaRigheNposizioniTest` falliranno).
- I Junit Test da superare per la consegna extra (facoltativa) sono quelli della classe **MatriceTraslaRigheNposizioniTest** (oltre a quelli della classe `MatriceTraslaRigheTest` che devono comunque essere superati).
- 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: Visita a serpentina di una matrice

Estratto e adattato da Compito II Appello del 27/06/2023.

Sia M una matrice di interi di dimensione $m \times n$, con $m > 1$ (quindi composta da almeno 2 righe) e $n > 0$. Gli elementi che appartengono alla serpentina della matrice M sono quelli che si incontrano partendo dalla cella di indice $[0][0]$, visitando gli elementi della matrice lungo le due diagonali (\searrow e \nearrow) come mostrato nei seguenti esempi, fino a che non si raggiunge l'ultima colonna della matrice. Nota: quando si arriva alla prima o all'ultima riga, si cambia la direzione della diagonale (si "rimbalza").

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32


(esempio 1)



Elementi che appartengono alla serpentina nell'esempio 1: $[1, 10, 19, 28, 21, 14, 7, 16]$

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

(esempio 2)



Elementi che appartengono alla serpentina nell'esempio 2: $[1, 5, 9]$

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

(esempio 3)



Elementi che appartengono alla serpentina nell'esempio 3: $[1, 9, 3, 11, 5, 13, 7]$

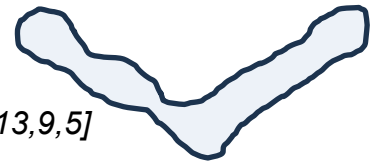
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

(esempio 4)



Elementi che appartengono alla serpentina nell'esempio 4: $[1, 9, 17, 25, 19, 13, 7]$

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



Elementi che appartengono alla serpentina nell'esempio 5: [1,7,13,9,5]

1	2	3
4	5	6

(esempio 6)



Elementi che appartengono alla serpentina nell'esempio 6: [1,5,3]

[CONSEGNA STANDARD]:

Scrivere un metodo Java-- di nome `visitaSerpentina` che, dati in input una matrice M di interi di dimensione $m \times n$, con $m > 1$ e $n > 0$, restituisca l'array di interi generato dalla visita a serpentina della matrice come precedentemente descritto.

[CONSEGNA CON DIFFICOLTÀ RIDOTTA]: Si può svolgere l'esercizio assumendo che la matrice M abbia un numero di colonne n uguale al doppio del numero di righe m meno uno, ovvero si può assumere che la matrice M abbia la seguente dimensione: $m \times (2m-1)$. In altre parole, si assuma che ci sia sempre un solo cambio di direzione della diagonale (vedere esempi 4, 5, 6).

NOTA BENE:

- Saranno premiate le soluzioni che occuperanno meno memoria dati.
- I Junit Test da superare per la consegna con difficoltà ridotta sono quelli della classe **SerpentinaRidottaTest** (quelli della classe `SerpentinaTest` falliranno).
- I Junit Test da superare per la consegna standard sono quelli della classe **SerpentinaTest** (oltre a quelli della classe `SerpentinaRidottaTest` che devono comunque essere superati).

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.