

– Simulazione di Prova Intermedia – 19/01/2021

Questa è una simulazione di prova intermedia.

Nello svolgimento degli esercizi si invitano gli studenti a considerare aspetti di “buona programmazione”, “pulizia del codice” ed efficienza. Ad es.: formattazione corretta del codice, rendere il codice modulare aggiungendo ove necessario altri metodi rispetto a quelli richiesti dall'esercizio, soprattutto se questi rendono il codice più pulito e leggibile, o se evitano duplicazione di codice. Inoltre, non ci devono essere warning nel codice scritto.

Esercizio Java-- n. 1: Test di Bilanciamento Array

Un array r di numeri interi si dice bilanciato se la sua lunghezza è un numero pari maggiore di zero, e se la somma del primo e dell'ultimo elemento dell'array è uguale alla somma del secondo e del penultimo elemento ed è anche uguale alla somma del terzo e del terzultimo elemento, se esistono, e così via. Un array composto da due soli elementi è bilanciato se i due numeri sono uguali. Un array vuoto (senza elementi) è non bilanciato.

Ad esempio:

- L'array $r=[-4, -4]$ è bilanciato, poiché l'array è composto di due soli elementi e i due numeri interi sono uguali.
- L'array $r=[12, -3, 22, 7]$ è bilanciato, poiché $12+7=-3+22$.
- L'array $r=[-3, 4, 100, -13, 9, -104, -8, -1]$ è bilanciato, poiché $-3-1=4-8=100-104=-13+9$.
- L'array $r=[24]$ è non bilanciato, poiché la sua lunghezza è un numero dispari.
- L'array $r=[24, 1, 24]$ è non bilanciato, poiché la sua lunghezza è un numero dispari.
- L'array $r=[0, 2, 2, 0]$ è non bilanciato, poiché $0+0 \neq 2+2$.
- L'array $r=[]$ (array vuoto) è non bilanciato, poiché non contiene nessun elemento.

Scrivere un metodo Java-- chiamato `testBilanciamento` che, dato in input un array r di numeri interi, restituisca `true` se r è bilanciato, e `false` altrimenti.

Nota bene: Gli studenti sono invitati a prediligere una soluzione ricorsiva. Una soluzione non ricorsiva (in cui si utilizzano istruzioni di iterazione, come cicli `for` o `while`) sarà comunque accettata e valutata dai docenti, ma la sua valutazione sarà penalizzata rispetto ad una soluzione ricorsiva.

Esercizio Java-- n. 2: Esplosioni a Catena

Sia T una matrice contenente numeri interi di valore uguale a 0 o a 1, di dimensione $m \times n$ (m righe e n colonne, con $m \geq 2$ e $n \geq 2$), e siano r e c due numeri interi tali che $0 \leq r \leq m-1$ e $0 \leq c \leq n-1$.

Definizione di intorno:

Si definisce intorno di $[r, c]$ l'insieme delle celle adiacenti ad $[r, c]$ (vedi le due figure di seguito: gli intorni sono le celle colorate).

Intorno di $[r, c] = [0, 0]$

c	0	1	2	3	4
r					
0					
1					
2					
3					

Intorno di $[r, c] = [1, 3]$

c	0	1	2	3	4
r					
0					
1					
2					
3					

Definizione di cella esplosiva successiva:

Se esiste, si definisce la cella esplosiva successiva a $[r, c]$ come la cella di indice $[x, y]$ in cui gli indici x e y vengono calcolati come segue:

- x è il massimo fra gli indici riga i delle celle $T[i, j]$ tali che:
 - $[i, j]$ appartiene all'intorno di $[r, c]$ e
 - $T[i, j] = 1$;
- fissato x come descritto prima, y è il massimo fra gli indici colonna j delle celle $T[x, j]$ tali che:
 - $[x, j]$ appartiene all'intorno di $[r, c]$ e
 - $T[x, j] = 1$.

La cella esplosiva successiva a $[r, c]$ non esiste se nell'intorno di $[r, c]$ non esistono celle contenenti il valore 1 (ovvero $T[i, j] = 0$ per tutte le celle nell'intorno di $[r, c]$).

Ad esempio, sia T la seguente matrice:

c	0	1	2	3	4	5	6	7	8
r	0	1	2	3	4	5	6	7	8
0	1	1	1	0	0	1	1	0	1
1	1	0	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1	0	1
3	0	0	1	0	1	0	1	0	0

- La cella esplosiva successiva a $[r, c] = [0, 0]$ corrisponde a quella di indice $[1, 0]$;
- La cella esplosiva successiva a $[r, c] = [3, 4]$ corrisponde a quella di indice $[2, 4]$.
- La cella esplosiva successiva a $[r, c] = [2, 8]$ non esiste, poiché tutte le celle nel suo intorno contengono valori uguali a 0.

Definizione di esplosione a catena:

Una esplosione a catena della matrice T a partire dalla cella di indice $[r, c]$ modifica la stessa matrice nel modo seguente:

- Se $T[r, c] = 0$, non si ha alcuna esplosione e la matrice non viene modificata.
- Se $T[r, c] = 1$, si eseguono i seguenti passi:
 - Si azzerà (esplode) la cella di indice $[r, c]$, ovvero $T[r, c] = 0$;
 - Si innesci una esplosione a catena a partire dalla *cella esplosiva successiva* a $[r, c]$, se questa esiste. Se tale cella non esiste allora la catena di esplosioni si interrompe e la matrice non viene ulteriormente modificata.

Ad esempio, sia T la seguente matrice:

1	0	0	0	1
1	1	1	0	0
0	0	0	0	1
1	0	0	0	1

L'esplosione a catena della matrice T a partire dalla cella $[r, c] = [0,0]$ modifica la stessa matrice come segue (step-by-step):

step1: azzero la cella $[0,0]$. L'indice della cella esplosiva successiva è $[1,1]$:

0	0	0	0	1
1	1	1	0	0
0	0	0	0	1
1	0	0	0	1

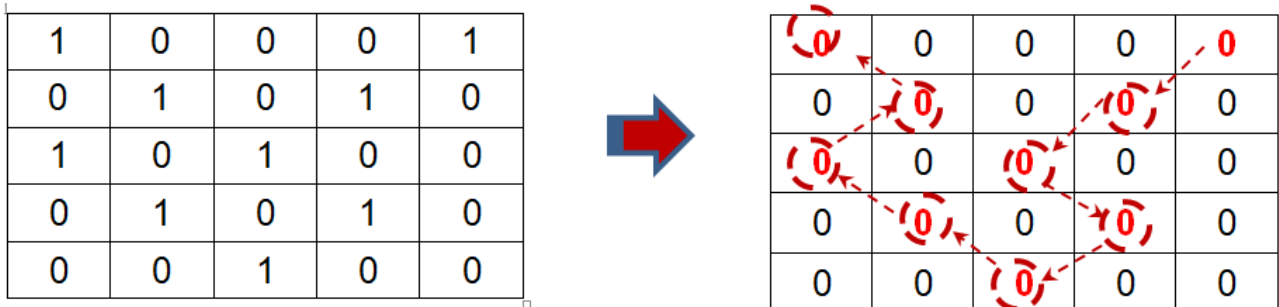
step2: azzero la cella $[1,1]$. L'indice della cella esplosiva successiva è $[1,2]$:

0	0	0	0	1
1	0	1	0	0
0	0	0	0	1
1	0	0	0	1

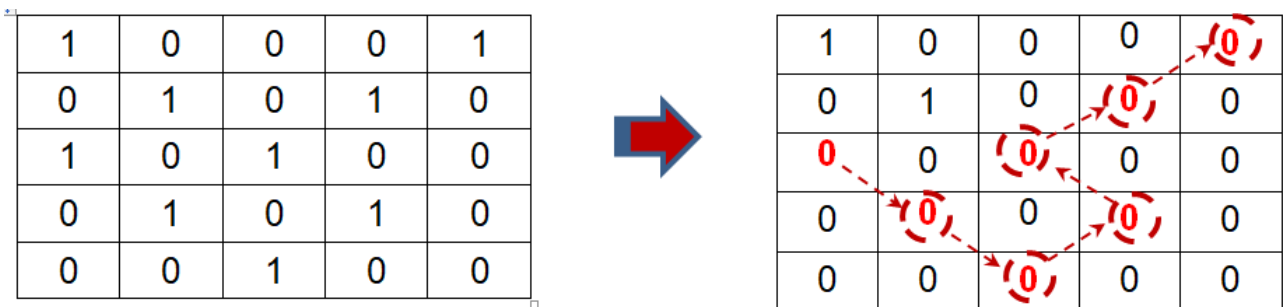
step3: azzero la cella $[1,2]$. Non esiste nessuna cella esplosiva successiva quindi la catene delle esplosioni si interrompe:

0	0	0	0	1
1	0	0	0	0
0	0	0	0	1
1	0	0	0	1

Altro esempio di esplosione a catena. A sinistra la matrice T di partenza, a destra la matrice al termine delle esplosioni a catena, con $[r, c] = [0, 4]$.



Altro esempio di esplosione a catena. A sinistra la matrice T di partenza, a destra la matrice al termine delle esplosioni a catena, con $[r, c] = [2, 0]$.



Consegna esercizio:

Scrivere un metodo Java-- chiamato **esplodiMatrice** che, dato in input una matrice T di numeri interi a valori 0 o 1 di dimensione $m \times n$ (m righe e n colonne), con $m \geq 2$ e $n \geq 2$, e dati due numeri interi r e c tali che $0 \leq r \leq m-1$ e

$0 \leq c \leq n-1$, restituisca la matrice T modificata applicandovi l'esplosione a catena a partire dalla cella di indice $[r, c]$.