

Corso di Algoritmi 2 - Esonero: parte di programmazione

Considerate un gioco tra due partecipanti P_0 e P_1 . Il gioco ha delle configurazioni istantanee ciascuna delle quali descrive lo stato del gioco in un istante. Dalla configurazione c_1 si può raggiungere la configurazione c_2 se una mossa legittima del giocatore che ha il turno di gioco nella configurazione c_1 trasforma c_1 in c_2 (non dovete preoccuparvi di sapere chi ha il turno di gioco, è tutto descritto nelle configurazioni). Esistono configurazioni vincenti per il giocatore P_0 e configurazioni vincenti per il giocatore P_1 ; un giocatore vince se il gioco entra in una configurazione vincente per quel giocatore.

Scrivete la classe Java **Partita** con

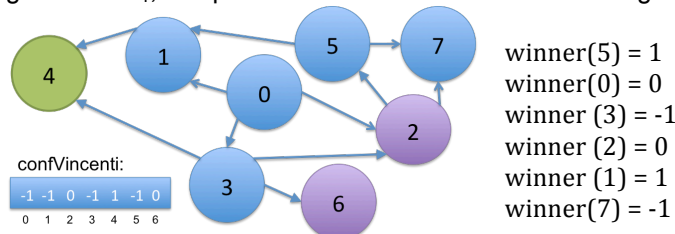
il costruttore **public Partita**(DirectedGraph gioco, int[] configVincenti)

dove: *gioco* è un grafo orientato, non pesato, in cui i nodi rappresentano configurazioni di gioco, ed esiste un arco (c_1, c_2) se dalla configurazione c_1 si può raggiungere la configurazione c_2 in una mossa; *configVincenti* specifica se una configurazione è vincente per un giocatore: $\text{configVincenti}[c] = -1$ se c non è vincente per nessuno, 0 se c è una configurazione vincente per P_0 , e 1 se c è una configurazione vincente per P_1 ;

e il metodo **public int winner**(int conf)

che restituisce 0 se a partire dalla configurazione di gioco *conf* la più vicina configurazione vincente è vincente per il giocatore 0 ; 1 se a partire dalla configurazione di gioco *conf* la più vicina configurazione vincente è vincente per il giocatore 1 ; -1 se sono vere le due condizioni sopra contemporaneamente oppure se nessuna configurazione vincente è raggiungibile da *conf*. Se *conf* non è una configurazione di gioco, il metodo solleva un'eccezione `java.lang.IllegalArgumentException`.

Esempio: dato il grafo in figura e configurazioni vincenti 2 e 6 per il giocatore P_0 , e 4 per il giocatore P_1 , l'output della funzione *winner* è come segue:



TEST: Lavorate implementando anche una classe test JUnit con almeno due test **significativi**, va bene anche se sono semplici. La classe test si deve chiamare **TestPartita**.

Struttura del progetto Java e consegna:

Le classi **Partita** e **TestPartita** devono essere contenute in un package il cui nome è il **vostro cognome** (in lettere minuscole). Consegnate l'intero package.

Suggerimenti e commenti:

- 1) Potete usare la libreria `graphLib.jar` e potete aggiungere tutti i metodi privati che volete
- 2) Suggerimento: utilizzate l'algoritmo più opportuno e tenete memoria in una apposita struttura delle configurazioni vincenti che l'algoritmo trova, nell'ordine in cui le trova; dopo, utilizzando questi e gli altri dati raccolti, determinate l'output del metodo *winner*
- 2) In uno dei test chiamate *winner* con input diversi, *sullo stesso oggetto*; è utile per verificare che sia tutto inizializzato correttamente
- 3) Selezionate la cartella "Consegna" come vostro workspace; in questo modo, in caso di problemi tecnici, il vostro lavoro non andrà completamente perduto, e comunque non rischiate di dimenticarvi di consegnare (se invece dovete decidere di ritirarvi, basta dirlo).
- 4) Verranno valutati: l'algoritmo utilizzato, la correttezza e l'aderenza alle specifiche (comprese la gestione degli input particolari e l'inizializzazione), la presenza di una classe test con test significativi.