

Corso di Algoritmi 2 - Prova pratica di programmazione - 22 giugno 2015

Considerate una interrete con n router e collegamenti tra i router.

Scrivete una classe Java **InterNetwork**

con il costruttore:

public InterNetwork(UndirectedGraph net)

dove net è un grafo non orientato e non pesato in cui

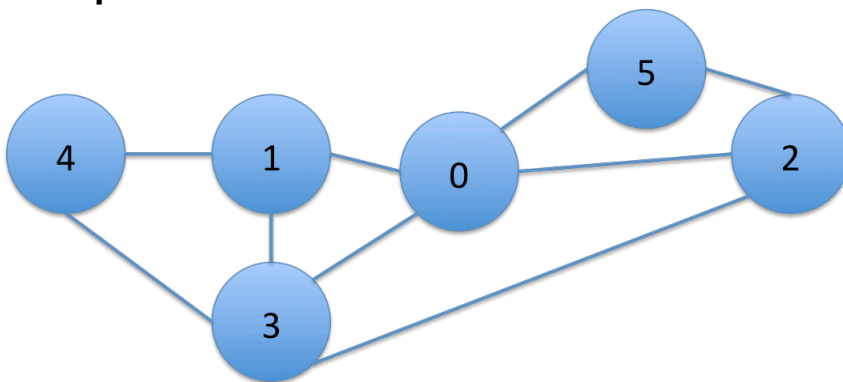
- i nodi rappresentano router,
- ciascun arco (r,t) è il collegamento tra il router r e il router t ,

e il metodo:

public int numberOfHops(int $r1$, int $r2$)

che restituisce il minimo numero di hop necessari per inoltrare un pacchetto da una rete locale connessa in rete tramite $r1$, ad una rete locale connessa in rete tramite $r2$, *contando anche $r1$ ed $r2$* (se $r1$ e/o $r2$ non sono nodi del grafo, restituisce -1).

Esempio:



numberOfHops(3,1) = 2
numberOfHops(5,4) = 4
numberOfHops(4,2) = 3
numberOfHops(1,1) = 1
numberOfHops(1,6) = -1
numberOfHops(7,-3) = -1

TEST: Naturalmente lavorate implementando anche una classe test JUnit con almeno due test significativi, va bene anche se sono semplici.

La classe di test si deve chiamare **TestInterNetwork**.

Suggerimenti e commenti:

- 1) Potete usare la libreria graphLib.jar
- 2) Potete aggiungere tutti i metodi private che volete.
- 3) Suggerimento per una possibile implementazione: il metodo numberOfHops chiama un metodo private che fa la visita opportuna; poi numberOfHops restituisce il valore appropriato che è stato calcolato durante la visita.
- 4) In uno dei test chiamate numberOfHops con input diversi, *sullo stesso oggetto*; non è obbligatorio, ma è utile per verificare che sia tutto inizializzato correttamente.

Consegna:

Consegnate la classe **InterNetwork.java** e la classe **TestInterNetwork.java**.