

Corso di Algoritmi 2 - Prova pratica di programmazione

L'azienda per il turismo di un'area montana commissiona un software per organizzare gite in montagna di diversi giorni, con soste nei rifugi. Si vuole un software che permetta di personalizzare le gite in base alle capacità dei partecipanti. Si richiede una prima versione del software che, tenendo conto della distanza che i partecipanti sono disposti a percorrere prima di fermarsi in un rifugio, proponga una suddivisione della zona nel minimo numero possibile di gite.

Scrivete una classe Java **Hikes** con il costruttore:

public Hikes(UndirectedGraph *rifugi*), dove *rifugi* rappresenta una mappa i cui nodi sono rifugi e i cui archi sono distanze tra due rifugi;
e il metodo:

public int zones(int *maxDistanza*)

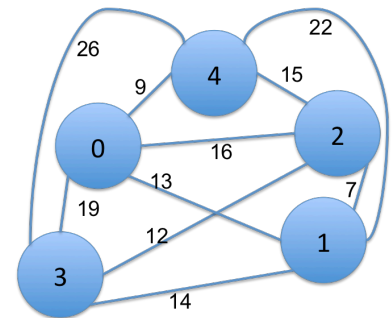
che restituisce il minimo numero di zone diverse in cui deve essere divisa la regione affinché, in ciascuna zona, ogni rifugio disti da almeno un altro rifugio non più di *maxDistanza*; il metodo restituisce una `java.lang.IllegalArgumentException` se *maxDistanza* è un numero negativo o nullo, oppure se le distanze tra due rifugi sono numeri negativi o nulli.

Esempio :

se la mappa *rifugi* è come in figura, **hikes(10) = 3**; **hikes(12) = 2**;

hikes(20) = 1;

hikes(-4) throws `java.lang.IllegalArgumentException`



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

Struttura del progetto Java e consegna:

Le classe richiesta e la classe test devono essere contenute in un package il cui nome è **il vostro nomecognome** (senza punti né spazi). Consegnate l'intero package.

Suggerimenti e commenti generali per tutti i progetti:

0) Dovete risolvere il problema rigorosamente applicando uno degli algoritmi visti a lezione.

1) Se serve, potete usare la libreria `graphLib.jar` (che sarà comunque sempre fornita; è allegato un file con le istruzioni per visualizzare i commenti Javadoc per i metodi) e consultare la documentazione delle API Java su `master.edu-al.unipmn.it`

2) Potete aggiungere tutti i metodi private che volete.

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 doveste 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.