

Prova d'esame del 12/09/2018 – Turno A

Si consideri il database “poweroutages”, contenente informazioni relative ai blackout che si sono verificati dal 2000 al 2014 negli USA. Il database, composto da 7 tabelle, ha la struttura schematizzata in Figura 1. La tabella ‘PowerOutages’ contiene le informazioni sulla data di inizio e di fine di ciascun blackout ed una stima delle persone coinvolte.

Si intende costruire un’applicazione JavaFX che permetta di interrogare tale base dati. L’applicazione dovrà svolgere le seguenti funzioni:

PUNTO 1

- Facendo click sul bottone *Crea Grafo*, creare un grafo semplice, pesato e non orientato, i cui vertici siano tutti i NERC presenti nella tabella ‘Nerc’. Gli archi del grafo devono rappresentare le coppie di NERC confinanti, secondo i confini contenuti nella tabella ‘NercRelations’.
- Il peso di un arco va calcolato come **il numero di mesi distinti** (tenendo conto degli anni) in cui c’è stata almeno un’interruzione di corrente in ciascuno dei due NERC considerati. Intuitivamente, il peso è un indice di **correlazione** delle interruzioni di corrente tra due NERC.

Ad esempio, considerando i dati riportati nella seguente tabella, gli eventi relativi alla coppia di NERC (PJM,ERCOT) con **id** 0 e 3 sono entrambi avvenuti a giugno 2015 (anche se in giorni diversi). Quindi il mese di Giugno 2015 contribuirà con un valore pari ad 1 al peso dell’arco. Qualora si verificassero altri eventi nello stesso mese in PJM-ERCOT, essi conteranno sempre 1 (infatti ciò che conta è il mese). Se si verificano altri eventi nei due NERC in un altro mese (ad esempio Maggio 2014), allora si sommerà un ulteriore contributo pari ad 1.

A scanso di equivoci, si evidenzia che Gennaio 2001 e Gennaio 2002 sono da considerarsi come mesi distinti (e così via: occorre considerare la coppia anno-mese).

NOTA: ricordare le funzioni SQL YEAR() e MONTH().

id	nerc	date_event_began	date_event_finished
0	PJM	2015-06-30 23:20:00	2015-07-12 17:00:00
3	ERCOT	2015-06-19 00:00:00	2015-06-20 18:00:00

- Popolare il menù a tendina con la lista dei NERC (i vertici del grafo). Al click del bottone *Visualizza Vicini*, stampare per il NERC selezionato la lista dei vicini ed il peso dell’arco che li collega, ordinati per peso decrescente.

PUNTO 2

- Effettuare una simulazione di tutte le interruzioni di corrente, per verificare se il modello proposto per la prevenzione delle catastrofi energetiche sia efficace: per garantire la continuità di servizio, ad ogni interruzione di corrente il NERC affetto chiede ad uno dei NERC vicini di coprire il fabbisogno energetico.
- L’utente inserisce nella casella di testo un numero di mesi **K**, e fa click il bottone *SIMULA*.
- Per ogni interruzione di corrente, creare due eventi, quello di inizio e di fine interruzione. Gli eventi sono ordinati cronologicamente. Alla ricezione del primo (inizio interruzione), il NERC affetto **x** chiede aiuto ad uno dei NERC vicini, utilizzando due criteri per scegliere il donatore, con la seguente priorità:
 - Il vicino a cui il NERC **x** ha donato energia **negli ultimi K mesi**.
 - Il vicino di **x** con l’indice di correlazione (peso dell’arco) minore (se il criterio 1. non trova alcun NERC).
 - Nel caso in cui il criterio 1. trovi più di un NERC “donatore” si scelga quello che soddisfa il criterio 2. tra i NERC trovati.

Un vincolo aggiuntivo è che ciascun NERC possa aiutare un solo altro NERC alla volta. Quindi i criteri precedenti devono considerare esclusivamente i NERC che in quell'istante non stanno donando energia. Alla ricezione del secondo evento (termine interruzione), viene registrato per il NERC donatore **un bonus** pari alla durata in giorni del disservizio.

- d. Nel caso in cui non ci sia un NERC vicino disponibile ad erogare la corrente, poiché tutti i vicini sono già impegnati ad aiutare altri NERC affetti da disservizio, si verifica una CATASTROFE.
- e. Al termine della simulazione stampare il totale dei bonus di ciascun NERC, ed il numero di catastrofi che si sono verificate.

Nella realizzazione del codice, si lavori a partire dalle classi (Bean e DAO, FXML) e dal database contenuti nel progetto di base. È ovviamente permesso aggiungere o modificare classi e metodi.

Tutti i possibili errori di immissione, validazione dati, accesso al database, ed algoritmici devono essere gestiti, non sono ammesse eccezioni generate dal programma.

