

# Prova d'esame del 12/07/2019 – Turno B

Si consideri il data-set "food\_pyramid\_mod" che riporta le qualità nutrizionali di una serie di cibi ed ingredienti. Il data-set (estratto dal sito <https://catalog.data.gov/dataset/mypyramid-food-raw-data-f9ed6>) contiene una tabella 'food' con vari tipi di cibi. Ciascun cibo può essere servito in porzioni di dimensione diversa (descritte nella tabella 'portion'), e ciascun cibo può includere zero o più ingredienti tra quelli riportati nella tabella 'condiment'. Gli ingredienti inclusi in ciascun cibo sono descritti dalla relazione 'food\_condiment'. Si noti che alcuni cibi particolarmente semplici (ad esempio: Whole milk, food\_code 11111000) possono comparire sia come cibo, che come ingrediente.

Si intende costruire un'applicazione JavaFX che permetta di interrogare tale base dati ed ottimizzare la propria dieta. L'applicazione dovrà svolgere le seguenti funzioni:

## PUNTO 1

- Si permetta all'utente di inserire un valore numerico intero (nella casella "Porzioni"), che identifica il numero di porzioni diverse (portion\_id) con cui quel cibo è rappresentato. Ad esempio, se Porzioni=1, si considerino tutti i cibi. Se Porzioni=3, si considerino i cibi che sono presenti con un numero di porzioni maggiore o uguale a 3.
- Alla pressione del tasto "Analisi", si costruisca un grafo semplice, pesato, orientato, i cui vertici rappresentino i cibi (food) identificati al punto precedente. Si popoli un menù a tendina con la descrizione di tali cibi.
- Gli archi devono essere inseriti nel modo seguente:
  - Gli archi rappresentano la "differenza di grassi": per ogni coppia di cibi, esiste un arco che va dal cibo con il valor medio di grassi maggiore (calcolato tenendo conto di tutte le sue porzioni, campo "saturated\_fats") verso il cibo con il valor medio di grassi minore.
  - Il peso degli archi è dato dalla differenza dei due valori medi. Se la differenza è pari a 0, l'arco non va inserito.
- Alla pressione del tasto "Grassi", si visualizzi l'elenco dei 5 cibi aventi la "differenza di grassi" minima tra quelli successivi al cibo selezionato nella tendina del punto b. Deve essere visualizzato il nome del cibo ed il numero di "Grassi" di differenza.

## PUNTO 2

- A partire dal grafo calcolato nel punto precedente, si permetta all'utente di avviare una simulazione del tempo necessario ad una cucina attrezzata per preparare una grande quantità di cibi.
- Si supponga che la cucina disponga di K (numero intero tra 1 e 10) stazioni di lavoro. Il valore di K è inserito dall'utente. Ciascuna stazione di lavoro può preparare un solo cibo per volta.
- Si supponga che, terminata la preparazione di un cibo in una stazione, il tempo necessario per preparare il cibo successivo sia pari al numero di "differenza di grassi" (tra il cibo corrente e quello successivo), interpretando il numero di grassi come valore espresso in ore.
- Nella fase iniziale, la simulazione viene avviata partendo dal cibo selezionato dall'utente (punto 1.b.): si predispongono le K stazioni di lavoro con K cibi diversi, successivi a quello iniziale, selezionati come i K cibi aventi le "differenze di grassi" minime, e prevedendo un tempo di preparazione come specificato al punto 2.c. Nota: è possibile che non tutte le stazioni possano operare, se non vi sono abbastanza cibi possibili.

- e. Ogni volta che una stazione di lavoro termina la preparazione di un cibo, per essa viene scelto il cibo adiacente a quello appena preparato che ha la “differenza di grassi” minima, purché tale cibo non sia ancora stato preparato né sia in preparazione in un’altra stazione.
- f. La simulazione termina quando non è più possibile trovare dei nuovi cibi da preparare e le stazioni sono tutte libere. Al termine della simulazione, si stampi il numero di cibi preparati ed il tempo totale di preparazione dei cibi.

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.**

