Esercitazione 5

Si vuole fornire una realizzazione del tipo di dato Dizionario mediante una tabella hash con gestione delle collisioni ad indirizzamento aperto con tecnica di scansione *hashing doppio*.

Si richiede di non ammettere l'uso di chiavi duplicate e di utilizzare
-m primo per dimensionare la tabella
-h1(key) = key mod m
-h2(key) = [key mod (m-1)] +1
Queste scelte di m, h1 e h2 garantiscono che, in caso di conflitto, la scansione esamini tutte le celle della tabella.

Esercitazione 5

Considerare il package Dizionario composto da:

- interfaccia Dizionario.java: la specifica del tipo di dato Dizionario
- interfaccia Hash.java: specifica alcuni metodi necessari per realizzare una tabella hash
- classe DizRecord.java: memorizza una coppia (key,elem)
- classe DizHashHD.java: realizzazione del Dizionario mediante tabella hash a indirizzamento aperto e hashing doppio

e completare l'implementazione della classe DizHashHD.java.
Consegnare l'intera classe DizHashHD.java

Consegnare entro Domenica 30 Novembre ore 23.55

Esercitazione 5: precisazioni

Rilevazione accessi alla tabella hash: il campo tot_accessi deve tener conto di tutti gli accessi alle celle della tabella hash da parte dei metodi insert, delete e search.

Metodo rehash(): per garantire che la sequenza di scansione esamini tutte le celle della tabella il metodo deve ridimensionare la tabella ad un numero primo. Allora, dopo aver moltiplicato la dimensione attuale per la costante RESIZE_FACTOR, è necessario determinare il più piccolo numero primo successivo e utilizzare quello come nuova dimensione della tabella. Per far ciò, utilizzare le seguenti istruzioni:

```
BigInteger prime = BigInteger.valueOf((long)(diz.length*Hash.RESIZE_FACTOR));
int newcapacity = (prime.nextProbablePrime()).intValue();
```

Esercitazione 5: precisazioni

Metodo insert(k,e): poiché non c'è completa garanzia di ottenere un numero primo dal metodo nextProbablePrime(), il metodo insert può fallire anche se c'e' spazio in tabella. Infatti può accadere che la sequenza di scansione esamini solo alcune celle e che queste siano tutte occupate. In tal caso il metodo insert deve tornare false.
Inoltre, bisogna tener conto sia dell'unicità delle chiavi presenti nel dizionario che delle eventuali cancellazioni effettuate. Ad esempio, partendo da una tabella vuota, inseriamo le chiavi x,y,z (intere non negative) in sequenza (prima x, poi y e poi z). Supponiamo che tutte e tre collidano nella stessa cella della tabella. Allora x prende il posto che le compete, y il successivo nella scansione e z il successivo ancora.

(continua…)

Esercitazione 5: precisazioni

0	1	2	3	4	5	6
x		y			z	

A questo punto cancelliamo y: allora la cella di y ha canc=true.

0	1	2	3	4	5	6
x		canc			z	

Infine, inseriamo di nuovo z: l'inserimento non deve andare a buon fine, ovvero la scansione non può fermarsi quando trova una cella con canc= true! La scansione può terminare quando trova una casella vuota (a null), oppure quando ha già esaminato tutte le celle della tabella.

(continua…)

Esercitazione 5: precisazioni

Nel caso la chiave non sia già presente in tabella, la posizione di inserimento è:
-quella della prima casella vuota, se non sono state attraversate celle cancellate;
-quella della prima cella cancellata attraversata, atrimenti.

Nella tabella precedente, se aggiungiamo una nuova chiave w che collide ancora con x,y,z (y gia' cancellata!) allora deve esserle assegnata la cella cancellata relativa ad y.

0	1	2	3	4	5	6
x		w			z	