

Esame di Programmazione II

Appello di giorno 05 Ottobre 2015 - Gruppo A
Università degli Studi di Catania - Corso di Laurea in Informatica

Testo della Prova

Definizione Iniziale.

Una *List-of-Trees* (LOT) di grado k è una struttura dati lineare implementata mediante una lista concatenata in cui i nodi sono degli alberi binari di ricerca contenuti al massimo k nodi. Gli elementi della struttura sono memorizzati nei nodi degli alberi binari che popolano la lista. Quando un nuovo elemento viene inserito in una LOT, esso viene inserito nell'albero che si trova in testa alla lista, a meno che quest'ultimo non contenga già k elementi. In tal caso l'inserimento verrà effettuato su di un nuovo albero vuoto, posizionato in testa alla lista. Quando viene effettuata la cancellazione di un elemento h , tutte le copie dell'elemento presenti nella LOT vengono cancellati. Se, dopo una cancellazione, uno degli alberi della LOT diventa vuoto, esso dovrà essere cancellato dalla lista.

La corretta implementazione di ciascuno dei seguenti esercizi permette l'acquisizione dei punti indicati. La corretta implementazione delle classi come template è facoltativa e permette l'acquisizione di ulteriori 3 punti.

1. **(5 punti)** Si fornisca una classe **C++**, denominata **List<H>**, che implementi una semplice lista doppiamente concatenata e non ordinata. La classe dovrà contenere almeno i seguenti metodi.
 - (a) **void insert(H *x)** è una procedura che inserisce il nuovo elemento puntato da x all'interno della lista.
 - (b) **void del(H *x)** è una procedura che elimina l'elemento puntato da x dalla lista. Nel caso in cui l'elemento non sia presente o sia nullo la procedura non ha effetto.
2. **(9 punti)** Si fornisca una classe **C++**, denominata **BST<H>** che implementi un albero binario di ricerca. La classe dovrà contenere i seguenti metodi.
 - (a) **void insert(H x)** è una procedura che inserisce un nuovo elemento x all'interno dell'albero.
 - (b) **void del(H x)** è una procedura che cancella l'elemento x dall'albero. Se l'elemento non è presente la procedura non ha effetto.
 - (c) **void print()** è una procedura che stampa gli elementi dell'albero utilizzando una visita inorder

3. **(13 punti)** Si fornisca una classe C++, denominata LOT<H> che rappresenti una List-of-Trees di grado k , così come definita sopra. Si forniscano le implementazioni delle seguenti procedure:

- (a) `void insert(H x)` è una procedura che inserisce un nuovo elemento x all'interno della struttura secondo le modalità definite sopra.
- (b) `void del(H x)` è una procedura che elimina tutte le copie dell'elemento x dalla struttura, se presente. Se la LOT non contiene elementi uguali a x allora la procedura non ha effetto.
- (c) `void print()` è una procedura che stampa gli elementi degli alberi contenuti nella lista partendo dall'albero di testa e procedendo in modo sequenziale fino all'albero di coda.

Si crei quindi un'istanza di LOT<int>, che rappresenti una LOT di grado 4, e si eseguano le seguenti operazioni:

```
insert(2), insert(3), insert(1), insert(5), insert(3), insert(7), insert(6),  
insert(1), insert(2), insert(4), insert(5), insert(3), insert(2), insert(1),  
del(5), del(1), del(2).
```

Si esegua in seguito la stampa della struttura. L'output sarà quindi:

```
3 4  
3 6 7  
3
```