

Esame di Programmazione II

Appello di giorno 8 Aprile 2015
Università degli Studi di Catania - Corso di Laurea in Informatica

Testo della Prova

Definizione Iniziale.

Una *Lista con accesso semi-diretto (SDList)* è una lista ordinata e doppiamente linkata a cui è associata un array di puntatori (chiamati *shortcuts*) che permettono di accedere più velocemente ad alcuni nodi della lista. Anche i puntatori contenuti all'interno dell'array associato alla lista sono ordinati, in ordine non decrescente, rispetto al valore contenuto nel nodo puntato dai relativi link.

Specifiche.

La corretta implementazione di ciascuno dei seguenti punti permette l'acquisizione di 9 punti. La corretta implementazione della classe come template è facoltativa e permette l'acquisizione di ulteriori 3 punti.

1. Si fornisca una classe C++, denominata `MySDList<H>`, che implementi la seguente interfaccia `SDList<H>`, che rappresenta una lista con accesso semi-diretto i cui elementi (che possono contenere ripetizioni) sono compresi tra 1 e 100. L'array di shortcuts associato alla lista dovrà contenere 10 puntatori, $\{p_0, p_1, \dots, p_9\}$, dove lo shortcut p_i punta al primo nodo della lista il cui elemento è maggiore di $10 \times i$, se presente, e punta a `null` altrimenti. La classe dovrà contenere i seguenti metodi virtuali.
 - (a) `SDList<H>* ins(H x)` aggiunge un nuovo elemento (nella corretta posizione) alla struttura dati e restituisce un puntatore ad un oggetto di tipo `SDList<H>`.
 - (b) `void print()` è una procedura che stampa in output gli elementi della lista, seguendo l'ordinamento della struttura.

Si crei quindi un'istanza di `MySDList<int>` e si inseriscano al suo interno i seguenti elementi:

7, 13, 2, 6, 9, 10, 21, 32, 4, 12

Si esegua in seguito la stampa dei valori inseriti nell'albero attraverso la procedura `print`. L'output del programma sarà quindi:

2, 4, 6, 7, 9, 10, 12, 13, 21, 32

```
template <class H> class SDList {
public:
    virtual SDList<H>* ins(H x) = 0;
    virtual void print() = 0;
}
```

2. i inserisca all'interno della classe `MySDList<H>` l'implementazione della seguente procedura `int search(H x)` che restituisce 1 se l'elemento `x` è presente nella struttura, 0 altrimenti.
3. Si inserisca all'interno della classe `MySDList<H>` l'implementazione della seguente procedura `MultiList<H>* del(H x)` che cancella un'occorrenza dell'elemento `x` dalla lista, se presente, e restituisce un puntatore ad un oggetto di tipo `MySDList<H>`. Si esegua in seguito la cancellazione, dall'istanza della struttura dati creata al passo precedente, degli elementi (nell'ordine) 7, 4, 32, 10. Si stampi in seguito il contenuto della lista. L'output del programma sarà quindi:
2,6,9,12,13,21