

Esame di Programmazione II

Appello di giorno 9 Settembre 2014
Università degli Studi di Catania - Corso di Laurea in Informatica

Testo della Prova

Definizione Iniziale.

Una *Lista di Array* (ArrayList) è una struttura dati lineare formata da una lista in cui ciascun nodo contiene un array di dimensione prefissata k . Gli elementi della struttura sono memorizzati all'interno degli array contenuti nei nodi della lista. I primi elementi vengono inseriti nell'unico nodo presente nella lista. Quando il numero di tali elementi eccede il valore k , un nuovo nodo della lista viene allocato per permettere la memorizzazione di ulteriori k elementi. Analogamente se l'array contenuto in un nodo della lista non contiene più elementi, allora esso viene eliminato dalla lista stessa. In ogni istante solo un nodo della lista (quello più a destra) potrà contenere posizioni vuote all'interno del proprio array.

Specifiche.

La corretta implementazione di ciascuno dei seguenti esercizi 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 `MyArrayList<H>`, che implementi la seguente interfaccia `ArrayList<H>`, che rappresenta una lista di array doppiamente linkata e non ordinata contenente i seguenti metodi virtuali.
 - (a) `ArrayList<H>* ins(H x)` aggiunge un nuovo elemento alla struttura dati e restituisce un puntatore ad un oggetto di tipo `ArrayList<H>`. L'inserimento avviene in coda;
 - (b) `ArrayList<H>* del(H x)` cancella l'elemento x dalla struttura dati, se presente, e restituisce un puntatore ad un oggetto di tipo `ArrayList<H>`;
 - (c) `int search(H x)` restituisce 1 se x è presente nella lista, 0 altrimenti;
 - (d) `void print()` è una procedura che stampa in output gli elementi della lista;Si crei quindi un'istanza di `MyArrayList<int>` in cui il valore di k sia pari a 5 e si inseriscano al suo interno i seguenti valori:

23 4 6 8 12 21 5 9 7 3 16 2 24

Si esegua poi la cancellazione dei valori 4, 5, 16 e 2.

Infine si esegua la stampa dei valori inseriti nella lista.

L'output del programma sarà quindi:

23 6 8 12 21 9 7 3 24

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

• • •

2. Si fornisca una classe derivata da `MyArrayList<H>`, denominata `OrderedArrayList<H>`, che rappresenti una lista di array ordinata. Si crei quindi un'istanza di `OrderedArrayList<int>` e si eseguano le stesse operazioni eseguite nel precedente punto. L'output del programma sarà quindi:

```
3 6 7 8 9 12 21 23 24
```

• • •

3. Si aggiunga alla classe `OrderedArrayList<H>` un nuovo metodo denominato `rank`, la cui definizione è quella data di seguito.

```
int rank(H x)
```

Tale metodo prende in input un elemento di tipo `H` e restituisce in output il suo rango, ossia la sua posizione all'interno dell'insieme ordinato contenuto all'interno della struttura dati. Tale valore corrisponde al numero di elementi più piccoli di x , aumentato di uno. Il metodo restituisce il valore 0 se il valore non è presente nella struttura dati.

Si esegua tre volte la procedura `rank` con input 3, 8 e 23, rispettivamente. L'output del programma sarà quindi:

```
1 4 8
```

• • •