

Cognome e Nome:
Numero di Matricola:

Docente:

Spazio riservato alla correzione

1	2	3	4	5	6	totale
/14	/10	/12	/14	/10	/30	/90

- [14 punti]** Aggiungere alla classe **BinarySearchTree**, che implementa l'interfaccia **Dictionary** mediante un albero binario di ricerca, il metodo
Entry min();
che restituisce la voce del dizionario con chiave più piccola. Il metodo deve essere implementato utilizzando solo i metodi delle interfacce **BinaryTree** e **Dictionary** ad esclusione dei metodi **elements**, **positions** ed **entries**.
- [10 punti]** Aggiungere alla classe **MyDictionary** che implementa l'interfaccia **Dictionary** il metodo
public Iterator removeAll(Object key) throws InvalidKeyException;
che rimuove tutte le voci del dizionario con chiave **key** e restituisce un iteratore delle voci rimosse. Il metodo deve essere implementato usando solo i metodi delle interfacce studiate.
- [12 punti]** Scrivere il metodo
public Entry insert(Object key, Object value) throws InvalidKeyException;
della classe che implementa una coda a priorità mediante un albero binario completo. Le voci della coda a priorità conoscono la loro posizione nell'albero.
- [14 punti]** Aggiungere alla classe **LinkedTree** il metodo
public Iterator leaves();
che restituisce un iteratore di tutti i nodi foglia dell'albero. Il metodo non deve invocare i metodi **elements** e **positions**.
- [10 punti]** Scrivere i seguenti metodi della classe **ListSet** che implementa l'interfaccia **Set** tramite una variabile di istanza di tipo **List**:
[7 punti] **public Set intersect(Set B);**
che restituisce l'intersezione tra B e l'insieme su cui è invocato
[3 punti] **public Object insert(Object element);**
che inserisce **element** nell'insieme restituendolo in **output**; se **element** appartiene già all'insieme non effettua l'inserimento e restituisce **null**.

Si assuma che la classe **ListSet** fornisca anche i metodi
 - public boolean contains(Object element)** che restituisce **true** se **element** appartiene all'insieme e **false** altrimenti
 - public Object remove()** che cancella un elemento a caso dall'insieme e lo restituisce.

6. [30 punti]

[20 punti] Provare il seguente teorema .

Se f è un flusso in una rete di flusso G con sorgente s e destinazione t , allora le seguenti condizioni sono equivalenti:

1. f è un massimo flusso in G
2. G_f non contiene augmenting path
3. $|f| = c(S,T)$ per un taglio (S,T)

[10 punti] Applicare l'algoritmo di Edmonds-Karp alla seguente rete di flusso. Illustrare graficamente tutti i passaggi dell'algoritmo (rete residua, cammino aumentante, rete di flusso).

