

ALBERI BINARI

Angelo Di Iorio

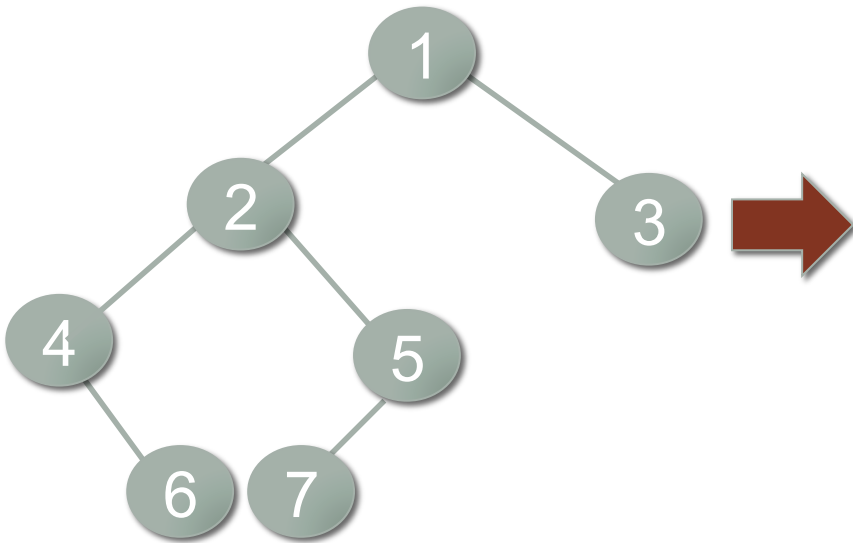
Università di Bologna

Esercizio 1

- Implementare una classe Java `BinaryTree` per memorizzare un albero binario di interi
- Realizzazione basata su puntatori
- La classe espone i metodi per:
 - inizializzare un nodo con un valore intero
 - aggiungere un nodo (sottoalbero) come figlio sinistro/destro di questo nodo
 - accedere ai nodi figli (sinistro/destro), al nodo padre e al valore intero
 - cancellare il figlio (sottoalbero) sinistro/destro

Esercizio 2

- Implementare una classe `BinaryTreeVisit` che espone un metodo per stampare a video l'albero binario, con una visualizzazione basic simile a quella usata per gli alberi (classe `TreeVisits` in esercitazione precedente)
- Esempio:



```
:1
  LeftChild:2
    LeftChild:4
      -
      RightChild:6
    RightChild:5
      LeftChild:7
      -
  RightChild:3
```

[semplice ma
ci accontentiamo]

Esercizio 3 (5.5 sul libro)

- Dato un albero binario t non vuoto, si vuole memorizzare, in ciascun nodo u , il numero di nodi che si trovano nel sottoalbero radicato in u .
- Si fornisca un algoritmo lineare nel numero di nodi.
- Si aggiunga alla classe `BinaryTree` un campo privato per memorizzare il numero di nodi e alla classe `BinaryTreeVisit` un metodo per settare questo valore su tutti i nodi di un albero passato in input

Esercizio 4 (5.6 sul libro)

- Dato un albero binario i cui nodi contengono interi, si vuole cancellare ogni foglia che sia un figlio sinistro e contenga lo stesso intero del padre.
- Aggiungere un metodo alla classe `BinaryTreeVisit`

Esercizio 5 (5.9 sul libro)

- Dato un albero binario, i cui nodi contengono elementi interi, si scriva una procedura di complessità ottima per ottenere l'albero inverso, ovvero un albero in cui il figlio destro (con relativo sottoalbero) è scambiato con il figlio sinistro (con relativo sottoalbero).
- Aggiungere un metodo alla classe `BinaryTreeVisit`

ALBERI BINARI DI RICERCA

Esercizio 6

- Implementare una classe Java per memorizzare un albero binario di ricerca, che contiene valori interi
- Realizzazione basata su puntatori
- La classe espone i metodi :
 - `BinarySearchTree(Integer i)`: inizializza uno nodo con un valore intero
 - `BinarySearchTree insert(Integer i)`: aggiunge un intero all'albero binario; ritorna il nodo inserito o `null` se il valore intero è già memorizzato nell'albero
 - `find(Integer i)`: cerca un intero nell'albero binario e ritorna l'elemento `BinarySearchTree` che lo contiene (`null` se non presente)

Esercizio 7

- Implementare un metodo che trova il valore minimo in un albero binario di ricerca (che contiene valori interi)
- Aggiungere il metodo alla classe `BinarySearchTree`

Esercizio 8

- Implementare un metodo che stampa i valori contenuti in un albero binario di interi in ordine decrescente
- Aggiungere il metodo alla classe `BinarySearchTree`