

Università di Pisa

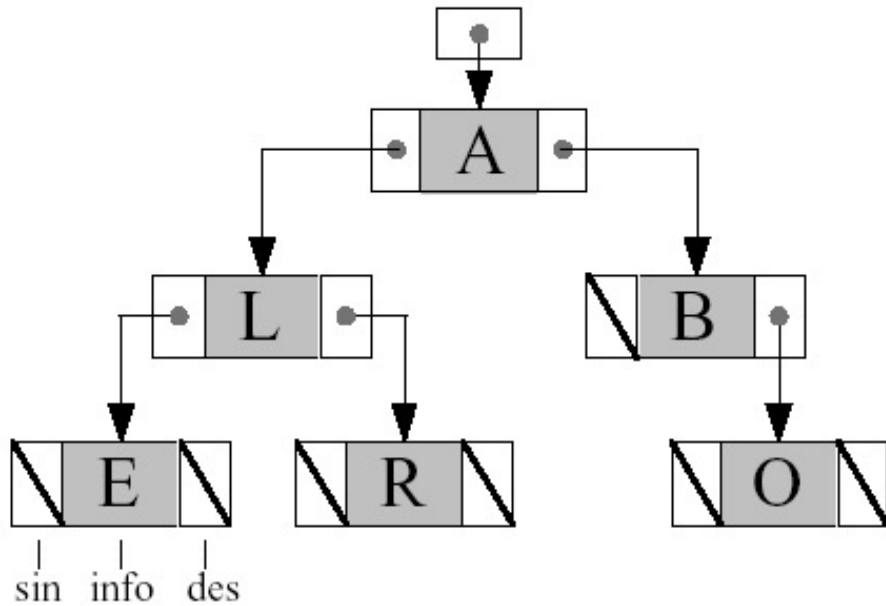
Pietro Ducange

Algoritmi e strutture dati

Alberi: Esercizi

a.a. 2020/2021

Esercizio 0



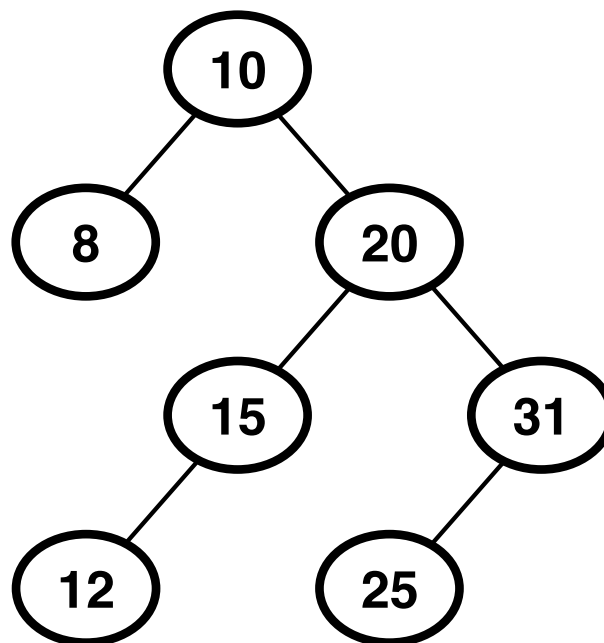
Visita Simmetrica: E L R A B O

Visita in Preordine: A L E R B O

Visita in Postordine: E R L O B A

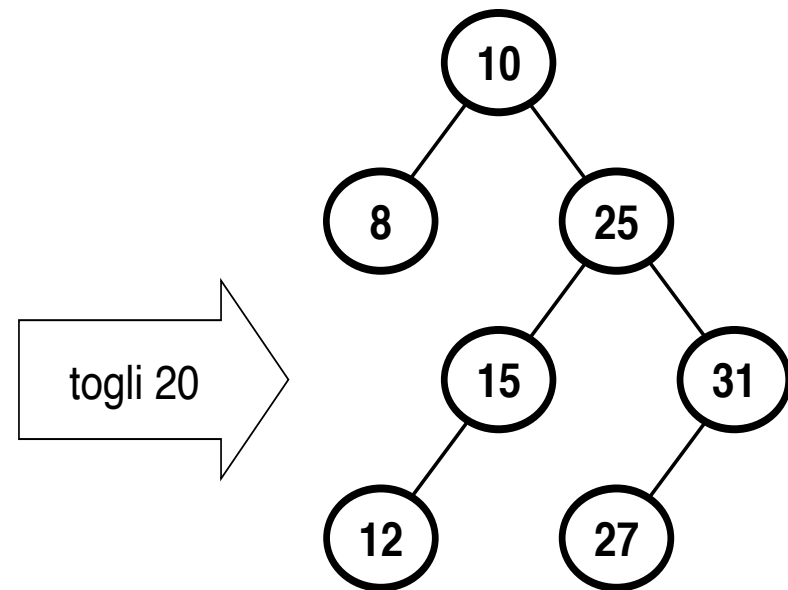
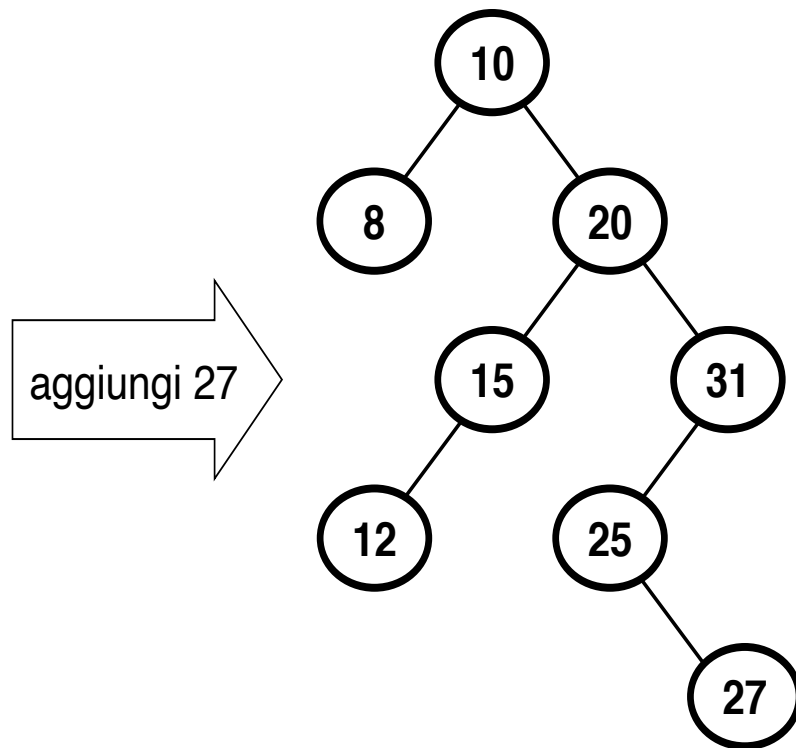
Esercizio 1

Dato il seguente albero binario di ricerca:



Disegnare gli alberi risultanti dopo l'aggiunta del valore 27 e la successiva eliminazione del valore 20.

Soluzione Esercizio1



Altri esercizi

1. Scrivere un programma C++ che dato un albero generico, somma 1 ad ogni sua etichetta
2. Scrivere un programma C++ che somma ad ogni nodo di un albero generico il numero dei suoi figli

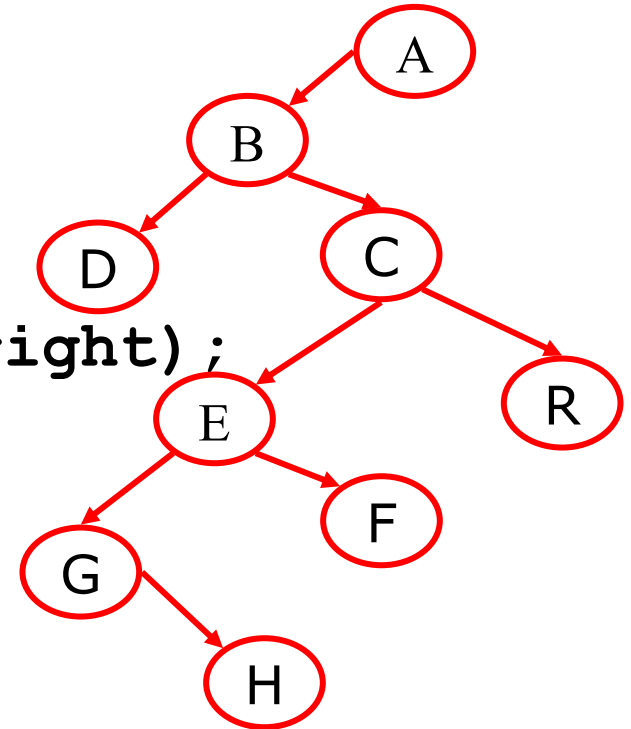
Soluzione Es1

Sommare 1 a ogni etichetta di un albero generico

```
void somma (Node *tree) {  
    if (! tree) return;  
    tree->label++;  
    somma (tree->left);  
    somma (tree->right);  
}
```

Soluzione Es 2: Sommare ad ogni nodo di un albero generico il numero dei suoi figli

```
int fratelli(Node* tree) {  
    if (!tree) return 0;  
    return 1+ fratelli(tree->right);  
}  
  
void somma(Node * tree){  
    if (! tree) return;  
    tree->label+= fratelli(tree->left);  
    somma(tree->left);  
    somma(tree->right);  
}
```



Esercizio I

Dato lo heap: [100, 90, 80, 70, 80, 50, 20, 10]

Indicare lo heap dopo una estrazione e il successivo inserimento del nodo 85. Indicare le chiamate a up e down

dopo una estrazione:

[90,80,80,70,10,50,20] down(0), down(1), down(4)

dopo l'inserimento di 85:

[90,85,80,80,10,50,20,70] up(7), up(3), up(1)

Esercizio II

Scrivere una funzione C++ che stampi in ordine simmetrico gli elementi di uno heap

```
void inorder(int * A, int i, int last){  
    if (i>last) return;  
    inorder(A, 2*i+1, last);  
    cout << A[i];  
    inorder(A, 2*i+2, last);  
}
```

Prima chiamata: `inorder (A, 0, last)`