

Gli alberi di ricerca binari

Anna Corazza

aa 2023/24

Dove studiare

- ▶ Sha'13, 5.4

Sha'13 Clifford A. Shaffer, Data Structures & Algorithm Analysis in C++, (edition 3.2), 2013

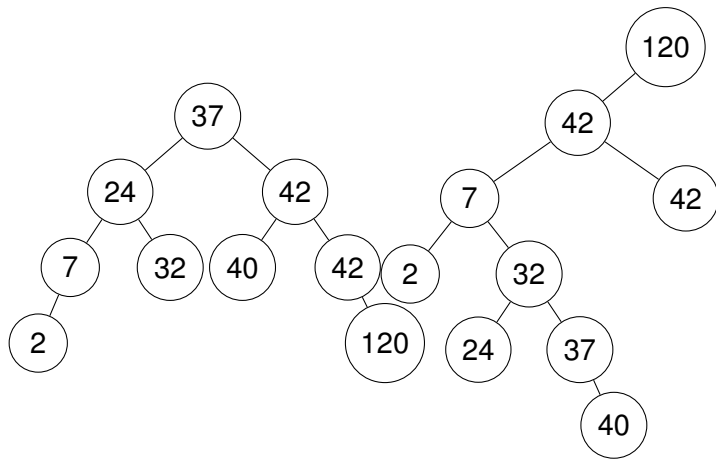
<https://people.cs.vt.edu/shaffer/Book/C++3elatest.pdf>

Definizioni e proprietà

- ▶ Strumento efficiente sia nella ricerca che nell'inserimento (i container lineari non lo sono)
- ▶ **Proprietà:** per ogni nodo con chiave K :
 - ▶ Chiavi nel sottoalbero sinistro $< K$.
 - ▶ Chiavi nel sottoalbero destro $\geq K$.
- ▶ Stampa con visita **inorder** va dalla chiave più piccola a quella più grande in modo ordinato.

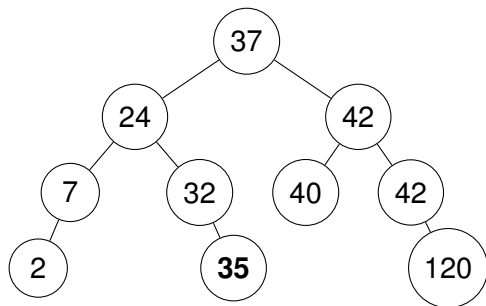
Esempi

2,7,24,32,37,40,42,42,120



Inserimento di un nodo

35



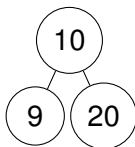
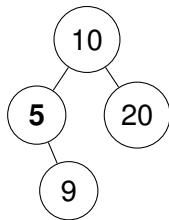
- ▶ Inserisco sempre una foglia: rischio di aumentare la profondità dell'albero.
- ▶ Per una ricerca efficiente, sarebbe meglio che l'albero fosse meno profondo possibile.

Rimuovi nodo con valore minimo

- ▶ Scendendo sempre a sin, il primo nodo che non a figlio sin.
- ▶ Per cancellare sposto il suo figlio destro al suo posto.
- ▶ Soluzione ricorsiva.

```
template <typename Key, typename E>
BSTNode<Key, E>* BST<Key, E>::
deletemin(BSTNode<Key, E>* rt) {
    if (rt->left() == NULL) // Found min
        return rt->right();
    else { // Continue left
        rt->setLeft(deletemin(rt->left()));
        return rt; }}

```

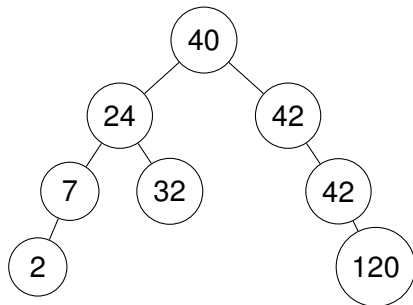
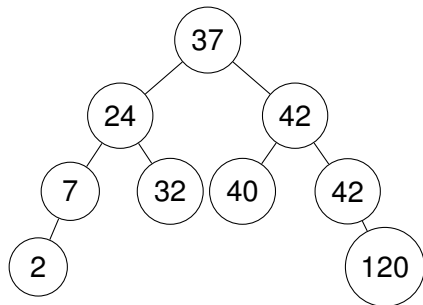


Rimuovere un nodo

- ▶ Prima trovarlo (R), poi rimuoverlo.
- ▶ Tre possibilità:
 - ▶ R non ha figli: metti a NULL il puntatore;
 - ▶ R ha un figlio: lo passi al puntatore
 - ▶ R ha due figli: due possibilità:
 - ▶ uno lo attacchi sopra, uno lo reinserisci: costoso
 - ▶ sostituire il valore di R con uno del sottoalbero (che poi verrà rimosso)
- ▶ la seconda è preferibile, ma quale nodo del sottoalbero utilizzare?
- ▶ deve continuare ad essere un BST e possibilmente non cambiare troppo la struttura.
 - ▶ il minimo del sottoalbero destro
 - ▶ il massimo del sottoalbero sinistro
- ▶ Nel caso in cui siano ammessi più nodi con lo stesso valore, devo per forza cercare il minimo del sottoalbero destro (che è un eventuale doppione).

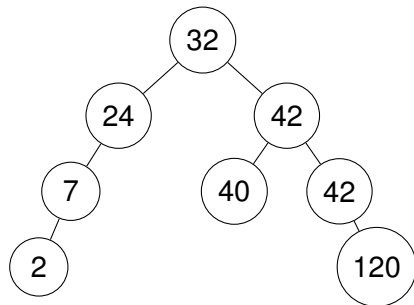
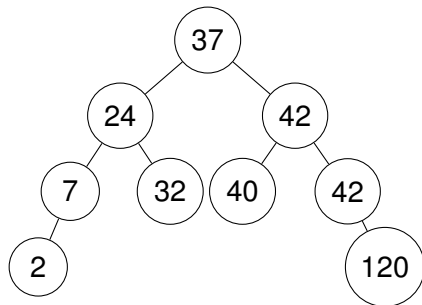
Esempio

Rimuovere 37



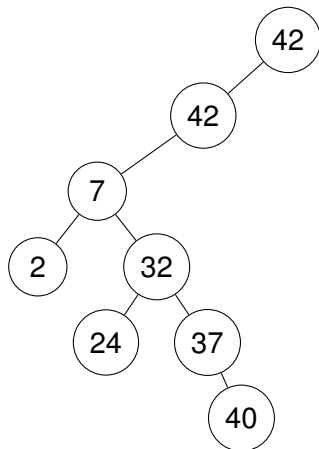
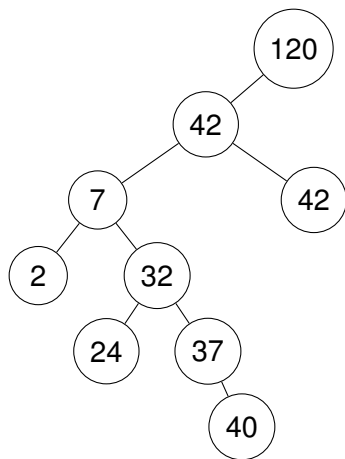
Esempio: alternativa

Rimuovere 37



Esempio: alternativa

Rimuovere 120



Non è più un albero
binario di ricerca!

Rimuovere un nodo

Conclusioni

- ▶ Se non c'è nulla in contrario, conviene cercare il minimo nel sottoalbero destro del nodo da rimuovere.
- ▶ Chiamiamo quindi `deletemin` e poi sostituiamo nel nodo il valore restituito.