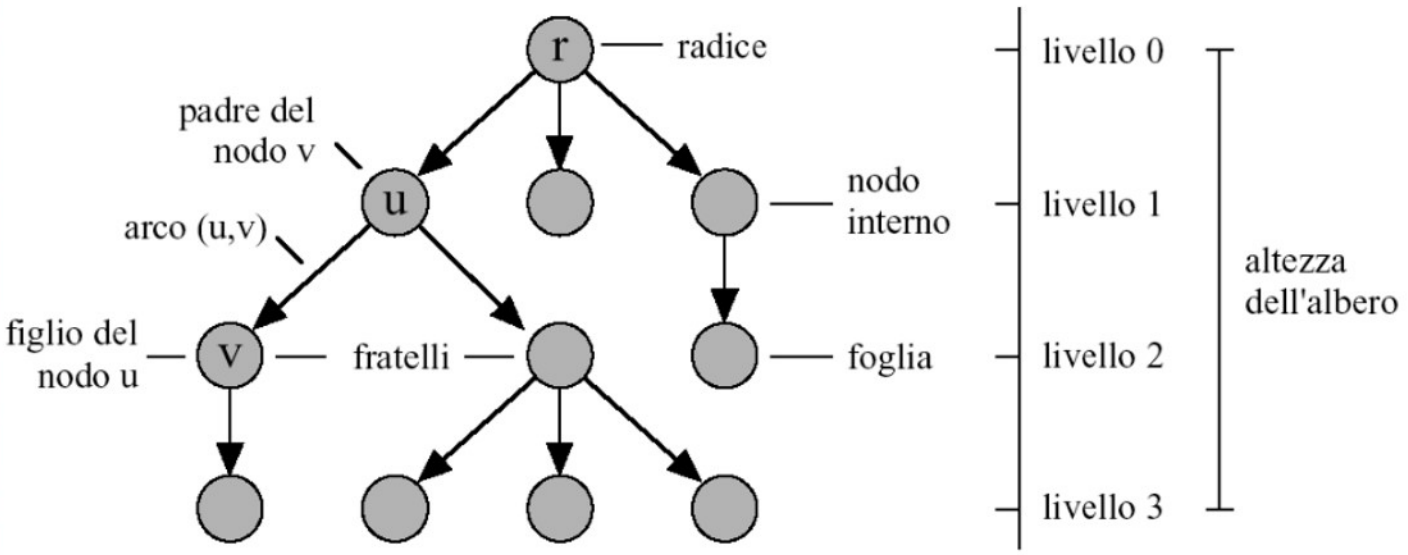
bst

domenica 23 giugno 2024 11:45

ALBERI



- Esistono alcune tipologie di alberi con vincoli strutturali che ne facilitano la rappresentazione o consentono di realizzare operazioni su collezioni di elementi in modo particolarmente efficiente
- I principali vincoli strutturali riguardano il grado dei nodi e la profondità
- un albero d-ario è un albero in cui i nodi hanno al più grado d
- un albero d-ario in cui $d=2$ si dice albero binario
- un albero binario completo è un albero binario in cui ogni livello è completamente pieno
- un albero binario quasi completo è un albero binario in cui ogni livello, tranne eventualmente l'ultimo, è completamente pieno, e tutti i nodi sono il più a sinistra possibile

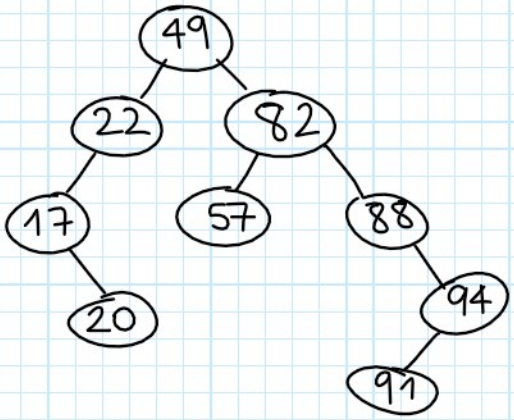
PROPRIETÀ 1: Un albero binario completo di altezza h ha $2^{h+1}-1$ nodi

PROPRIETÀ 2: Sia T un albero binario quasi completo di altezza h. Allora il numero n di nodi è tale che $2^h \leq n \leq 2^{h+1}-1$

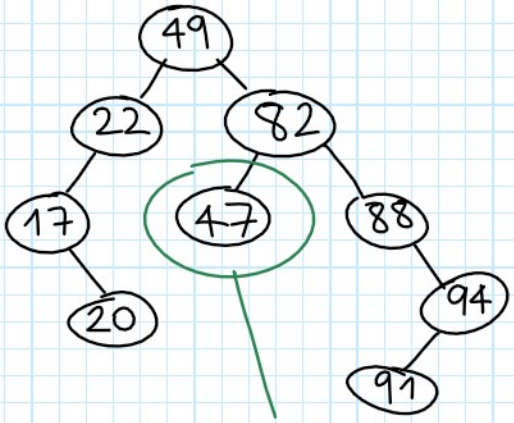
PROPRIETÀ 3: L'altezza di un albero binario quasi completo T con n nodi è $h = \lfloor \log_2 n \rfloor$ ($\lfloor \rfloor$ indica la parte intera)

ALBERI BINARI DI RICERCA (BST)

Sono alberi binari e non devono per forza essere completi o quasi completi.
Ad ogni nodo è associato un elemento ed una chiave presa da un dominio totalmente ordinato, quindi ogni chiave deve essere confrontabile con le altre.
Le chiavi degli elementi nel sottoalbero sinistro di un vertice V sono ≤ chiave(v) mentre gli elementi nel sottoalbero destro sono ≥ chiave(v)



albero binario di ricerca



albero binario NON di ricerca

domanda esame bst (aulaweb)

1)

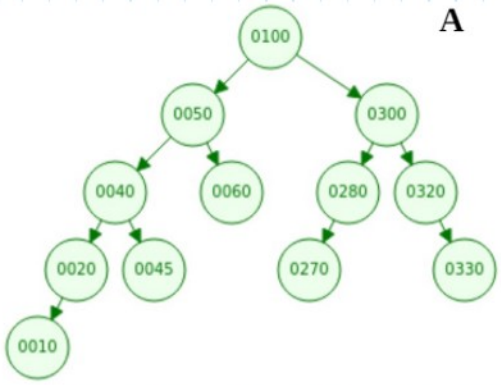
Si consideri il seguente BST che indicheremo con **A**.

[1/3 del punteggio] Assumendo che le chiavi siano numeri interi, si disegni come viene modificato il BST **A** dopo la seguente chiamata (senza fornire alcuna spiegazione dei passaggi: disegnate solo il risultato)

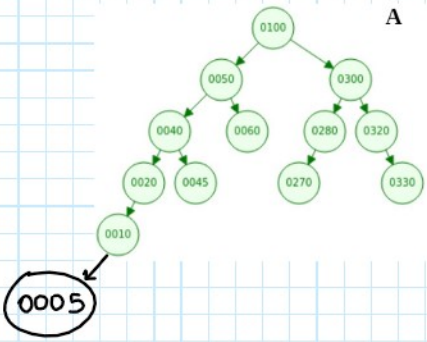
```
insertElem(5, "elem", A);
```

La chiamata `insertElem(5, "elem", A)` rappresenta il caso peggiore della operazione insertElem sui BST, rispetto alla complessità temporale? Motivare la risposta.

A



Si, rappresenta il caso peggiore perchè l'inserimento è in theta di h, con h l'altezza dell'albero. Per inserire 0005 si fa theta(h).



2)

[2/3 del punteggio] Si spieghino dettagliatamente, possibilmente mediante disegni chiari e autoesplicativi, i passaggi principali della chiamata

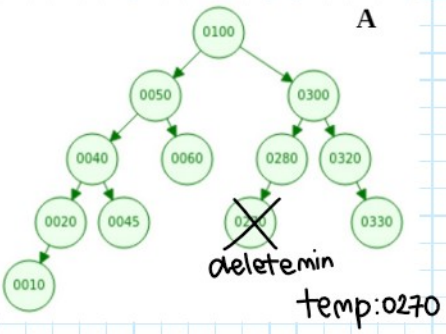
```
deleteElem(100, A);
```

effettuata sull'albero **A** modificato a seguito dell'inserimento dell'elemento con chiave `5`.

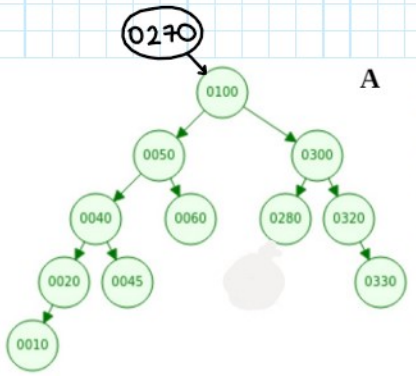
La chiamata `deleteElem(100, A)` rappresenta il caso peggiore della operazione deleteElem sui BST, rispetto alla complessità temporale? Motivare la risposta.

DeleteElem cancella un nodo, per trovarlo scorre tutto l'albero. Devo usare la funzione ausiliare deletemin che trova il nodo più piccolo. Troviamo il più piccolo dei grandi (a destra). Quindi in questo caso deletemin elimina 0270 e passa la sua chiave valore a deleteElem e deleteElem sostituisce il 0100 con 0270.

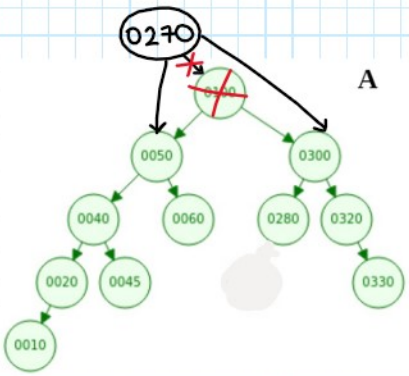
1)



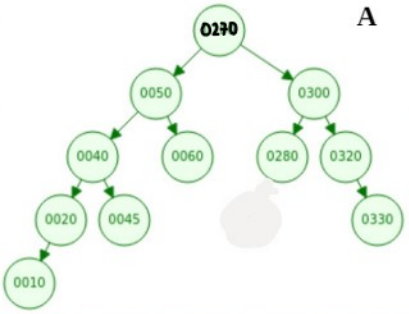
2)



3)



4)



complessità:

Tutte le operazioni su BST hanno un costo $O(h)$ dove h è l'altezza dell'albero binario di ricerca