Algoritmi e Strutture Dati

Stefania Monica
stefania.monica@unipr.it

Università degli Studi di Parma

A.A. 2019/2020

Alberi Binari di Ricerca

Definizione

Si dice albero binario di ricerca un albero binario tale che, per ogni nodo, se il nodo contiene una chiave di valore k

- Ogni nodo del suo sottoalbero sinistro contiene una chiave di valore minore o uguale a k
- Ogni nodo del suo sottoalbero destro contiene una chiave di valore maggiore o uguale a k

Osservazione

In un albero binario di ricerca, per implementare un algoritmo di ricerca non è necessario visitare tutti i nodi sistematicamente

Algoritmi per Alberi Binari di Ricerca

Algoritmi

- Ricerca
- Inserimento
- Rimozione

Algoritmi per Alberi Binari di Ricerca

Costo

- Nel caso ottimo, cioè quello in cui
  - Il valore cercato si trova nella radice
  - Il valore da inserire si trova nella radice
  - La rimozione deve essere fatta in un albero di ricerca vuoto

  non ci sono iterazioni, quindi

$$T(n) = \mathcal{O}(1)$$

Algoritmi per Alberi Binari di Ricerca

Costo

- Nel caso pessimo, cioè quello in cui bisogna compiere tutto il percorso dalla radice alla foglia più distante da essa prima di concludere l'operazione ricerca, inserimento o rimozione, il costo è proporzionale all'altezza h.
  Poiché l'altezza di un albero è massima quando l'albero degenera in una lista, in questo caso

$$T(n) = \mathcal{O}(n)$$

- Si può dimostrare che nel caso medio il costo è

$$T(n) = \mathcal{O}(\log(n))$$

Algoritmi per Alberi Binari di Ricerca

Osservazioni

- Il costo delle operazioni di ricerca, inserimento e rimozione in un albero binario di ricerca nel caso pessimo dipende dalla struttura dell'albero
- Per garantire che il costo nel caso pessimo sia $\mathcal{O}(\log(n))$, bisogna fare in modo che le foglie siano il più possibile allo stesso livello
  - Solo in questo caso si può garantire che l'altezza h dell'albero cresca come il logartimo del numero n di nodi

Algoritmi per Alberi Binari di Ricerca

Osservazioni

- Dal punto di vista dell'inserimento, occorre mettere i nuovi nodi nei livelli già esistenti così da riempirli il più possibile prima di creare nuovi livelli
- Dal punto di vista della rimozione, occorre ridistribuire i nodi rimasti all'interno dei livelli nel modo più uniforme possibile
- Inserimenti e rimozioni devono inoltre garantire che l'albero binario risultante sia ancora un albero binario di ricerca.

Alberi Bilanciati

Bilanciamento in Altezza

- Un albero si dice bilanciato in altezza se, per ogni nodo dell'albero, l'altezza del sottoalbero sinistro e l'altezza del sottoalbero destro differiscono di al più un'unità
- Gli alberi bilanciati in altezza sono anche detti alberi AVL (Adelson, Velsky, Landis)

Alberi Bilanciati

Fattore di Bilanciamento

- Il fattore di bilanciamento $\beta(n)$ di un nodo n è definito come la differenza tra l'altezza del sottoalbero sinistro e quella del sottoalbero destro.
- Indicando il sottoalbero sinistro di n con $sx(n)$ e il sottoalbero destro di n con $dx(n)$ si ha che

$$\beta(n) = h(sx(n)) - h(dx(n))$$

Osservazione

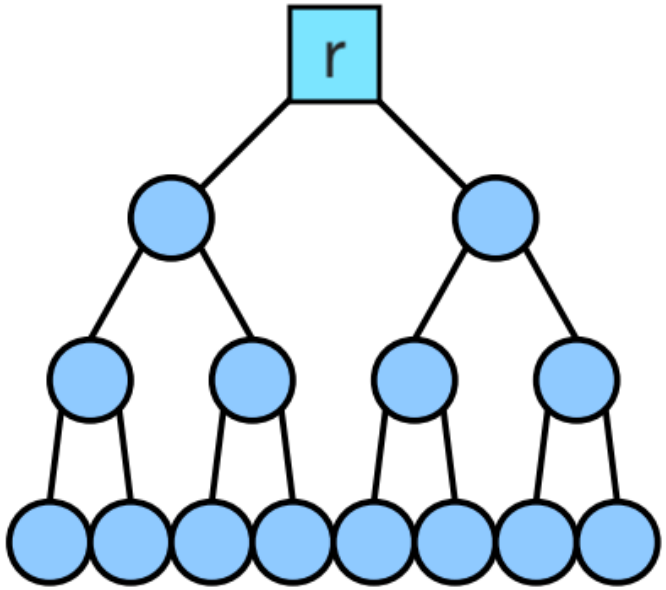
In base alle definizioni precedenti, un albero binario di ricerca è un albero AVL se, per ogni $n \in N$,

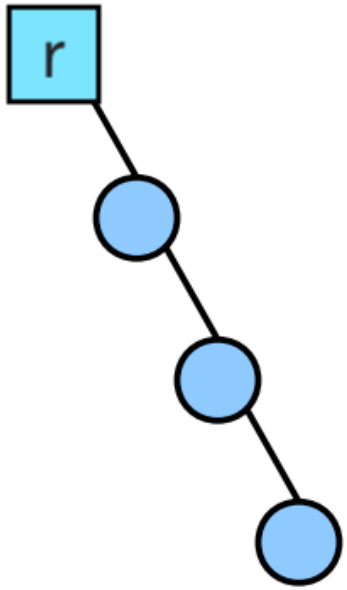
$$|\beta(n)| \leq 1$$

Alberi Bilanciati

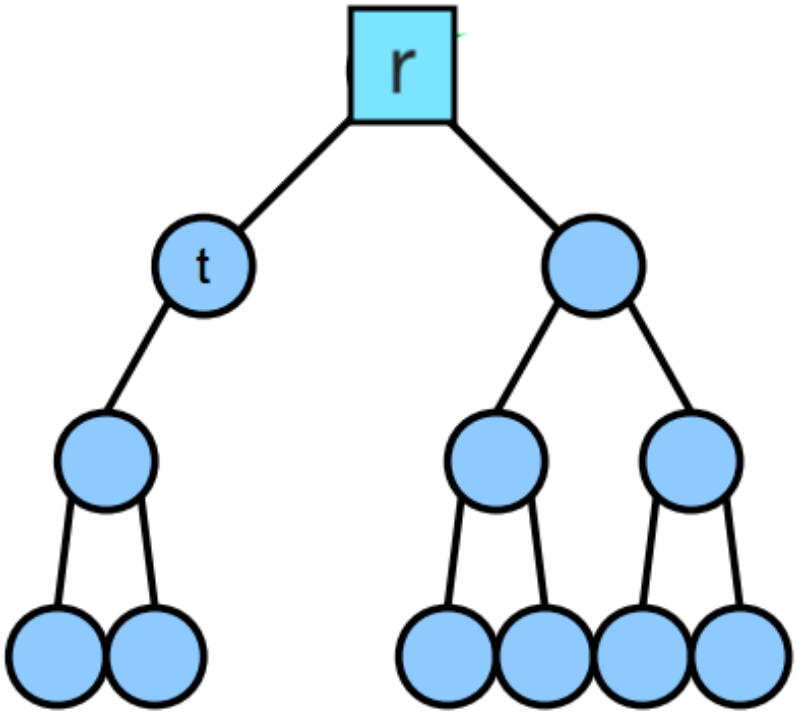
Esempi

- A parità di numero di nodi, il fattore di bilanciamento dei nodi di un albero binario di ricerca è massimo quando l'albero degenera in una lista.
- Il fattore di bilanciamento dei nodi di un albero binario di ricerca completo è 0.





Alberi Bilanciati



Osservazione

Per valutare l'altezza di un albero AVL, consideriamo gli alberi più sbilanciati che si possano costruire a parità di numero di nodi.

Alberi di Fibonacci

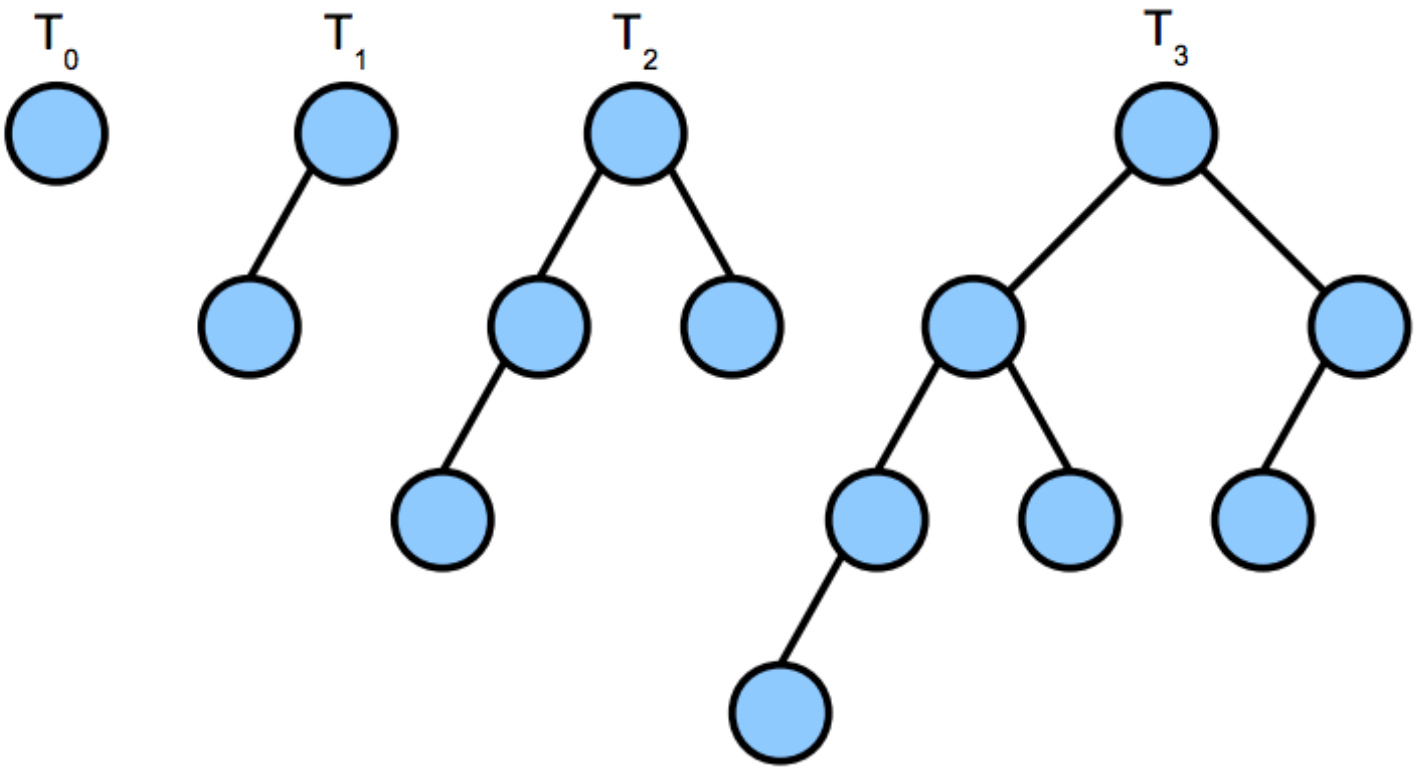
Alberi di Fibonacci

- Un albero di Fibonacci di altezza 0 è un albero con un unico nodo (radice)
- Un albero di Fibonacci di altezza 1 è un albero in cui la radice ha un unico figlio sinistro
- Un albero di Fibonacci di altezza $h > 1$ può essere costruito unendo, tramite l'aggiunta di un nodo radice, un albero di Fibonacci di altezza $h - 1$ e un albero di fibonacci di altezza $h - 2$

Alberi di Fibonacci

Alberi di Fibonacci - Esempi

Alberi di Fibonacci di altezza $h = 0$, $h = 1$, $h = 2$, $h = 3$



Alberi di Fibonacci

Osservazione

Tra tutti gli alberi di altezza h bilanciati in altezza, un albero di Fibonacci ha il minimo numero di nodi

Proposizione

Sia $\mathcal{T}_h$ un albero di Fibonacci di altezza h e sia n_h il numero dei suoi nodi. Si ha che

$$h = \Theta(\log(n_h))$$

Dimostrazione

- Per costruzione

$$n_h = 1 + n_{h-1} + n_{h-2}$$

Alberi di Fibonacci

Dimostrazione

- Si può dimostrare che induzione che

$$n_h = F_{h+3} - 1$$

  - Caso base: Se $h = 0$, $n_0 = 1 = F_3 - 1$
  - Ipotesi induttiva: Supponiamo che la tesi valga per ogni $k \in \{0, \ldots, h\}$, i.e.: $n_k = F_{k+3} - 1$
  - Dimostro che la tesi vale per $h + 1$, i.e.: $n_{h+1} = F_{h+4} - 1$

$$n_{h+1} = 1 + n_h + n_{h-1} = 1 + F_{h+3} - 1 + F_{h+2} - 1 = F_{h+4} - 1$$

- Ricordando che $F_h = \frac{1}{\sqrt{5}}(\phi^h - \hat{\phi}^h) = \Theta(\phi^h)$ si ottiene che

$$n_h = \Theta(\phi^{h+3}) = \Theta(\phi^h)$$

da cui

$$h = \Theta(\log(n_h))$$

Alberi Bilanciati

Corollario

Un albero AVL con n nodi ha altezza $\mathcal{O}(\log(n))$

Dimostrazione

- Sia h l'altezza dell'albero AVL considerato e sia n il suo numero di nodi
- Un albero di Fibonacci con altezza h ha n_h nodi, con $n_h \leq n$
- Ricordando che $h = \Theta(\log(n_h))$, i.e.:

$$\exists a > 0, b > 0, n \in \mathbb{N} \; t.c. \; a \cdot \log(n_h) \leq h \leq b \cdot \log(n_h)$$

si ha che

$$\exists b > 0, n \in \mathbb{N} \; t.c. \; h \leq b \cdot \log(n)$$

i.e. $h = \mathcal{O}(\log(n))$

Alberi Bilanciati

Osservazioni

- Il corollario precedente mostra che nel caso di alberi AVL gli algoritmi di ricerca, inserimento e rimozione visti per gli alberi binari di ricerca hanno costo $\mathcal{O}(\log(n))$ nel caso pessimo.
- Dobbiamo ancora vedere come mantenere il bilanciamento a seguito di inserimenti e rimozioni di nodi.