Algoritmi e Strutture Dati

Alberi AVL

Maria Rita Di Berardini, Emanuela Merelli[1]

[1]Dipartimento di Matematica e Informatica
Università di Camerino

Alberi AVL

Definizione (*bilanciamento in altezza*): un albero è bilanciato in altezza se le altezze dei sottoalberi sinistro e destro di ogni suo nodo differiscono al più di uno

Gli alberi AVL sono alberi binari di ricerca **bilanciati** in altezza

AVL (Adelson-Velskii e Landis)

Definizione (*fattore di bilanciamento*): il fattore di bilanciamento $\beta(v)$ di un nodo v è la differenza tra l'altezza del suo sottoalbero sinistro e quella del suo sottoalbero destro

$$\beta(v) = altezza[left[v]] - altezza[right[v]]$$

Alberi AVL

Il fattore di bilanciamento è tanto migliore quanto più piccolo è il suo valore assoluto

- alberi completi: il fattore di bilanciamento di ogni nodo è 0
- alberi bilanciati (alberi AVL): $|\beta(v)| \leq 1$ per ogni nodo v
- alberi degenerati in una lista: il fattore di bilanciamento è pari altezza dell'albero ($n - 1$ dove n è il numero di nodi dell'albero)

Definizione (*bilanciamento in altezza*) – definizione alternativa: un albero è bilanciato in altezza se, per ogni nodo v, $|\beta(v)| \leq 1$

Altezza di un albero AVL

Alberi bilanciati (e quindi alberi AVL) sono particolarmente adatti per realizzare efficienti strutture di tipo dizionario (insiemi dinamici su cui eseguiamo inserimenti, cancellazioni e ricerche)

L'altezza h di un albero T bilanciato è logaritmica nel numero di nodi n, ossia $h = \Theta(\log_2 n)$

Nota 1: $n \leq 2^{h+1} - 1$, cioè del numero di nodi di un albero binario completo di altezza h (fornisce un limite inferiore al valore di h)

Nota 2: $n \geq N(h)$, dove $N(h)$ denota il **minimo** numero di nodi che un albero bilanciato di altezza h **deve** avere (fornisce un limite superiore al valore di h)

Altezza di un albero AVL

Nota 1: $n \leq 2^{h+1} - 1$, ossia del numero di nodi di un albero binario completo di altezza h

$$n \leq 2^{h+1} - 1 \Longrightarrow n + 1 \leq 2^{h+1} \Longrightarrow \log_2(n+1) \leq h + 1$$

e quindi

$$h \geq \log_2(n+1) - 1$$

Altezza di un albero AVL

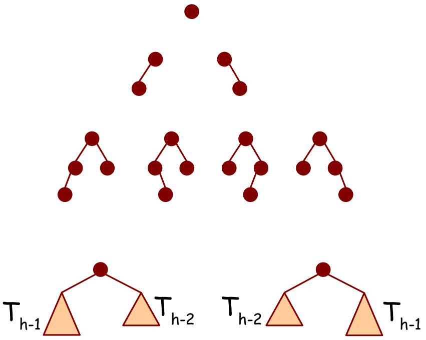
Nota 2: $n \geq N(h)$, dove $N(h)$ denota il **minimo** numero di nodi che un albero bilanciato di altezza h **deve** avere

Quanto vale $N(h)$ al variare di $h \geq 0$? Dobbiamo identificare la struttura di tutti i possibili alberi bilanciati di altezza h con minino numero di nodi (questi alberi vengono detti alberi di Fibonacci)

Nel seguito indichiamo con T_h un albero di Fibonacci di altezza h

Altezza di un albero AVL

altezza	Possibili strutture di T_h	N(h)
0		1
1		2
2		4
h		1 + N(h-1) + N(h-2)



Altezza di un albero AVL

Ricapitolando ...

$$N(h) = \begin{cases} 1 + N(h-1) + N(h-2) & \text{se } h \geq 2 \\ 1 & \text{se } h = 0 \\ 2 & \text{se } h = 1 \end{cases}$$

Ricorda molto da vicino i ...

Altezza di un albero AVL

Ricapitolando ...

$$N(h) = \begin{cases} 1 + N(h-1) + N(h-2) & \text{se } h \geq 2 \\ 1 & \text{se } h = 0 \\ 2 & \text{se } h = 1 \end{cases}$$

Ricorda molto da vicino i ... numeri di Fibonacci

$$F(n) = \begin{cases} F(n-1) + F(n-2) & \text{se } n \geq 3 \\ 1 & \text{se } n = 1, 2 \end{cases}$$

Possiamo dimostrare (per induzione su $h \geq 0$) che

$$N(h) = F(h+3) - 1$$

Altezza di un albero AVL

Possiamo dimostrare (per induzione su $h \geq 0$) che

$$N(h) = F(h+3) - 1$$

Caso Base

$h = 0$: $N(h) = 1 = 2 - 1 = F(3) - 1$
$h = 1$: $N(h) = 2 = 3 - 1 = F(4) - 1$

Passo induttivo sia $h > 1$. Allora:

$$\begin{array}{lcl} N(h) & = & N(h-1) + N(h-2) + 1 \\ \text{per ip. ind.} & = & (F(h-1+3) - 1) + (F(h-2+3) - 1) + 1 \\ & = & F(h+2) + F(h+1) - 1 \\ \text{per la def. di } F & = & F(h+3) - 1 \end{array}$$

Altezza di un albero AVL

Posto $\phi = \frac{1+\sqrt{5}}{2} \approx 1.618$ e $\hat{\phi} = \frac{1-\sqrt{5}}{2} \approx -0.618$ abbiamo che

$$F(n) = \frac{1}{\sqrt{5}}\left(\phi^n - \hat{\phi}^n\right)$$

Quindi

$$N(h) = \frac{1}{\sqrt{5}}\left(\phi^{h+3} - (\hat{\phi})^{h+3}\right) - 1 = \frac{1}{\sqrt{5}}\phi^{h+3} - \frac{1}{\sqrt{5}}(\hat{\phi})^{h+3} - 1$$

Altezza di un albero AVL

Distinguiamo i seguenti casi

- h pari e $h+3$ dispari:

$$-\frac{1}{\sqrt{5}}(\hat{\phi})^{h+3} = \frac{1}{\sqrt{5}}(0,618)^{h+3} \geq -1$$

- h dispari e $h+3$ pari:

$$-\frac{1}{\sqrt{5}}(\hat{\phi})^{h+3} = -\frac{1}{\sqrt{5}}(0,618)^{h+3} \geq -\frac{1}{\sqrt{5}}(1)^{h+3} = -\frac{1}{\sqrt{5}} \geq -1$$

Possiamo concludere che

$$N(h) = \frac{1}{\sqrt{5}}\phi^{h+3} - \frac{1}{\sqrt{5}}(\hat{\phi})^{h+3} - 1 \geq \frac{1}{\sqrt{5}}\phi^{h+3} - 2$$

Altezza di un albero AVL

Sia T una albero binario di altezza h ed n il numero dei suoi nodi;

$$n \geq N(h) \geq \frac{1}{\sqrt{5}}\phi^{h+3} - 2 \quad \text{e quindi} \quad n + 2 \geq \frac{1}{\sqrt{5}}\phi^{h+3}$$

Allora:

$$\begin{array}{lcl}
\log_2(n+2) & \geq & \log_2(\frac{1}{\sqrt{5}}\phi^{h+3}) \\
\quad \log ab = \log a + \log b & = & \log_2(\frac{1}{\sqrt{5}}) + \log_2(\phi^{h+3}) \\
\quad \log a^k = k \log a & = & \log_2(\frac{1}{\sqrt{5}}) + (h+3)\log_2(\phi) \\
 & = & \log_2(\frac{1}{\sqrt{5}}) + h\log_2(\phi) + 3\log_2(\phi) \\
\quad k \log a = \log a^k & = & h\log_2(\phi) + \log_2(\frac{1}{\sqrt{5}}) + \log_2(\phi^3) \\
\quad \log a + \log b = \log ab & = & h\log_2(\phi) + \log_2(\frac{1}{\sqrt{5}}\phi^3)
\end{array}$$

Altezza di un albero AVL

Ora:

$$\log_2(n+2) \quad \geq \quad h\log_2(\phi) + \log_2(\tfrac{1}{\sqrt{5}}\phi^3)$$

implica

$$h\log_2(\phi) \quad \leq \quad \log_2(n+2) - \log_2(\tfrac{1}{\sqrt{5}}\phi^3)$$

Infine, poichè $\phi = \frac{1+\sqrt{5}}{2} > 1$ implica $\log_2 \phi > 0$, abbiamo

$$h \quad \leq \quad a\log_2(n+2) + b$$

dove $a = \frac{1}{\log_2(\phi)}$ e $b = -\frac{\log_2(\frac{1}{\sqrt{5}}\phi^3)}{\log_2(\phi)}$

Rotazioni

Abbiamo visto come alberi binari di ricerca bilanciati consentono la ricerca di un dato elemento in un tempo logaritmico

Problema: l'inserimento o la cancellazione di un nodo potrebbero far perdere il bilanciamento di un albero AVL

È indispensabile mantenere il bilanciamento dell'albero anche se si inseriscono o si cancellano elementi

Il bilanciamento deve essere ripristinato mediante delle opportune operazioni sui nodi che prendono il nome di rotazioni

Rotazioni di base

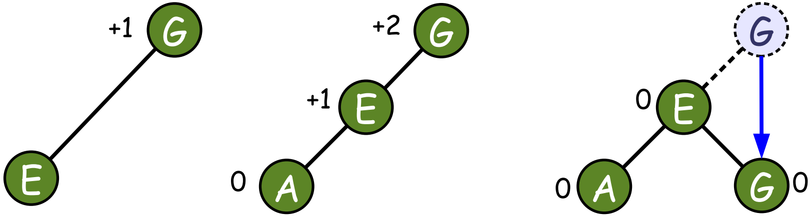
Esistono quattro tipi di rotazioni di base (semplici):

- **rotazioni SS**: inserimento a sinistra nel sottoalbero sinistro
- **rotazioni SD**: inserimento a destra nel sottoalbero sinistro
- **rotazioni DD**: inserimento a destra nel sottoalbero destro
- **rotazioni DS**: inserimento a sinistra nel sottoalbero destro

Queste rotazioni consistono nel portare l'elemento intermedio alla radice e nel far ridiscendere il nodo che causa lo sbilanciamento

Rotazioni di base

Caso semplice: il bilanciamento è perso a seguito dell'inserimento di un nodo in un albero costituito da due soli nodi

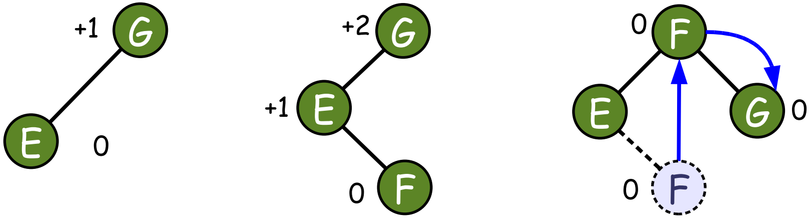


a) albero bilanciato

b) inserimento a sinistra del sottoalbero sinistro (sbilanciamento)

c) Rotazione SS

Rotazioni di base



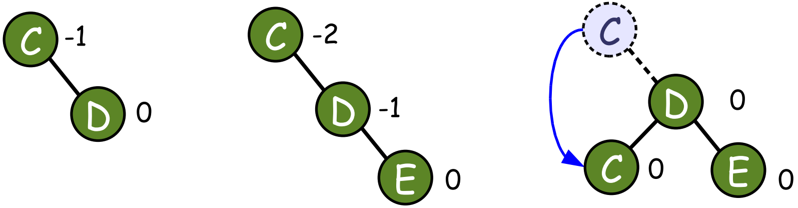
a) albero bilanciato

b) inserimento a destra del sottoalbero sinistro (sbilanciamento)

c) Rotazione SD

Rotazioni di base

Si possono eliminare eventuali sbilanciamenti simmetrici usando le rotazioni DD e DS

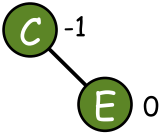


a) albero bilanciato

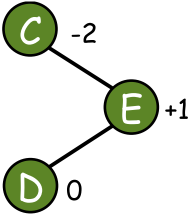
b) inserimento a destra del sottoalbero destro (sbilanciamento)

c) Rotazione DD

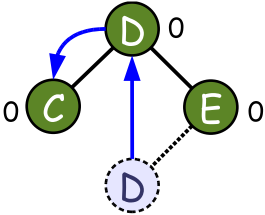
Rotazioni di base



a) albero bilanciato



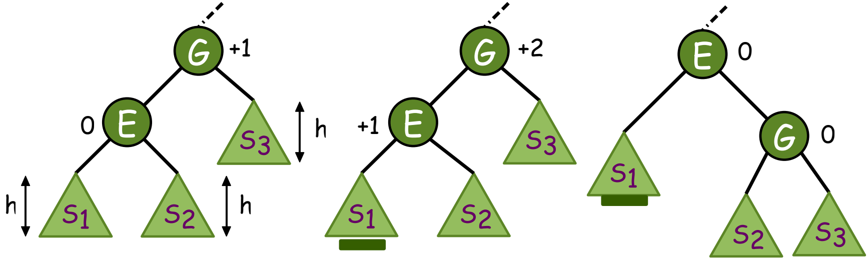
b) inserimento a sinistra del sottoalbero destro (sbilanciamento)



c) Rotazione DS

In generale ...

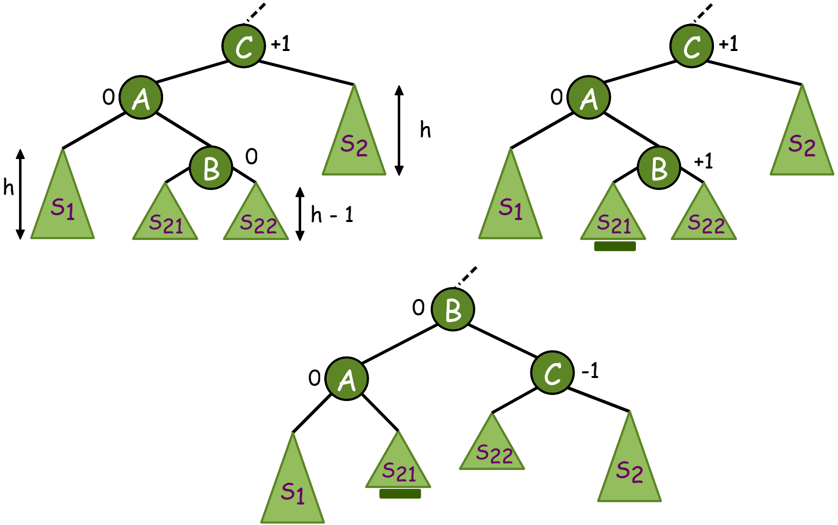
Lo sbilanciamento (ed il conseguente ribilanciamento) si possono verificare alla radice di un sottoalbero di un albero più grande

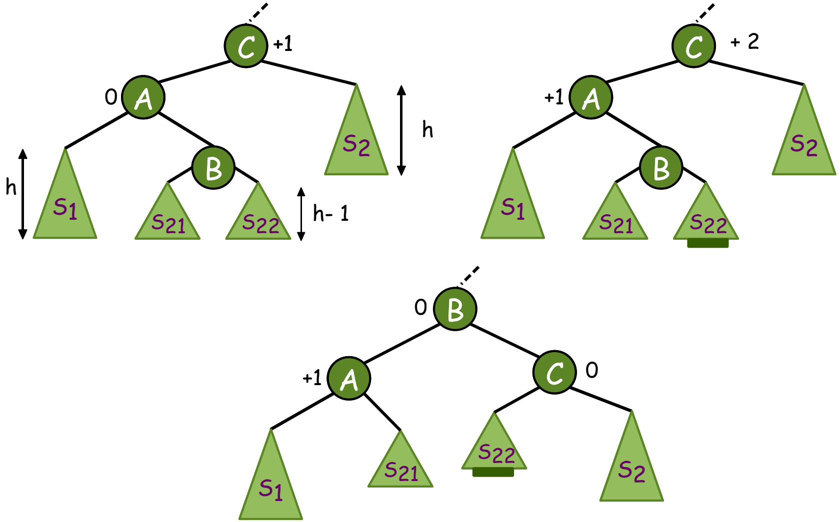


a) albero bilanciato

b) inserimento di (sbilanciamento)

c) Rotazione SS


C
+1
0
A
B
0
S2
h
h
S1
S21
S22
h - 1
C
+1
0
A
B
+1
S2
S1
S21
S22
0
B
0
A
C
-1
S1
S21
S22
S2


C
+1
0
A
B
S2
h
h
S1
S21
S22
h- 1
C
+ 2
+1
A
S2
B
S1
S21
S22
0
B
+1
A
C
0
S22
S1
S21
S2

Inserimento di un elemento in un albero AVL

(1) con il solito metodo di ricerca localizziamo la posizione in cui inserire il nuovo elemento; durante tale operazione memorizzato il puntatore al più basso nodo v il cui fattore di bilanciamento è -1 o $+1$

(2) viene effettuato l'inserimento

(3) vengono aggiornati i fattori di bilanciamento dei nodi lungo il percorso tra il nodo appena inserito e v; tutti i nodi lungo questo percorso avevano un fattore di bilanciamento 0 e vengono portati a $+1$ o -1; l'unico nodo che potrebbe registrare uno sbilanciamento è proprio v (**nodo critico**)

Inserimento di un elemento in un albero AVL

(4) se il fattore di bilanciamento di v è $+1$ (-1) ed il nodo è stato inserito nel sottoalbero destro (sinistro) il sottoalbero con radice v è automaticamente bilanciato; in caso contrario, il sottoalbero è sbilanciato ed richiede un ribilanciamento

(5) viene determinata ed effettuata la rotazione necessaria

Tutti questi passi, rotazioni comprese, interessano solo i nodi lungo il cammino dalla radice alla nuova foglia inserita

Il costo dell'inserimento di un elemento è proporzionale all'altezza dell'albero e, poichè l'albero viene mantenuto bilanciato, tale costo è logaritmico

Cancellazione di un elemento in un albero AVL

(1) si effettua la cancellazione di un elemento come descritto dall'algoritmo **Tree-Delete**

(2) si ricalcolano i fattori di bilanciamento mutati in seguito alla cancellazione; osserviamo che i sol fattori di bilanciamento che possono cambiare sono quelli dei nodi lungo il cammino dalla radice al nodo eliminato e che questi possono essere ricalcolati risalendo l'albero dal basso verso l'alto

(3) eseguiamo una rotazione per ogni nodo il cui fattore di bilanciamento è ± 2

Cancellazione di un elemento in un albero AVL