Alberi RB: $h \leq 2 * log_2(n+1) \in O(log_2(n))$

Alessandro Ferro

April 22, 2022

Abstract

Si vuole dimostrare che l'altezza di un albero Red-Black è un O - grande del logaritmo in base 2 del numero di nodi. In particolare l'altezza è sempre minore o uguale a $2 * log_2(n+1)$

1 Dimostrazione

Sia
· h(x) l'altezza dell'albero radicata nel nodo x;
· bh(x) [black height], il numero di nodi neri lungo qualche percorso dal nodo x a una foglia. Viene escluso il colore di x stesso. Questo numero viene indicato come "altezza nera".
· NNI(x) il numero di nodi interni (ovvero il numero di nodi non foglia) nell'albero radicato nel nodo x.

1.1 Dimostrazione NNI(x) $\geq 2^{bh(x)} - 1$

Prima di procedere a dimostrare la tesi principale, dobbiamo prima dimostrare che il numero di nodi interni radicati nel nodo x è maggiore o uguale a $2^{bh(x)-1} - 1$

Dimostrazione 1 Dimostriamo questo per induzione sull'altezza dell'albero radicato nel nodo x.

Caso base h(x) = 0 Se l'altezza radicata nel nodo x è 0, allora x è una foglia nera *null*. L'altezza nera radicata in x è 0 [$bh(x) = 0$], così come il numero di nodi interni [NNI(x) = 0]. Otteniamo dunque

$$NNI(x) \geq 2^{bh(x)} - 1$$
$$NNI(x) \geq 0$$

dunque è quindi valido per il caso base.

Caso induttivo h(x) ≥ 1 Se l'altezza dell'albero radicata in x è maggiore o uguale di 1, allora x è un nodo interno con 2 figli.
Sappiamo che l'altezza di un albero è sicuramente maggiore dell'altezza dei suoi sottoalberi, ma non possiamo dire lo stesso dell'altezza **nera**.
Infatti, supponiamo che y sia un nodo figlio di x.

$y.color = RED \implies bh(x) = bh(y);$
$y.color = BLACK \implies bh(x) = bh(y) + 1$

Allora, $bh(y) \geq bh(x) - 1$.
Sia z un altro figlio di x: vale lo stesso ragionamento fatto con y, allora

$$\begin{cases} bh(y) \geq bh(x) - 1 \\ bh(z) \geq bh(x) - 1 \end{cases}$$

Poiché la funzione esponenziale è una funzione monotona, possiamo scrivere

$$\iff \begin{cases} 2^{bh(y)} \geq 2^{bh(x)-1} \\ 2^{bh(z)} \geq 2^{bh(x)-1} \end{cases}$$

Sottraiamo 1 ambo i membri:

$$\iff \begin{cases} 2^{bh(y)} - 1 \geq 2^{bh(x)-1} - 1 \\ 2^{bh(z)} - 1 \geq 2^{bh(x)-1} - 1 \end{cases}$$

Poiché gli alberi radicati in y e z hanno un'altezza inferiore all'albero radicato in x, per ipotesi induttiva possiamo scrivere

$$\begin{cases} NNI(y) \geq 2^{bh(y)} - 1 \\ NNI(z) \geq 2^{bh(z)} - 1 \end{cases}$$

Ma allora

$$\begin{cases} NNI(y) \geq 2^{bh(y)} - 1 \geq 2^{bh(x)-1} - 1 \\ NNI(z) \geq 2^{bh(z)} - 1 \geq 2^{bh(x)-1} - 1 \end{cases}$$

Quindi per transitività otteniamo

$$\begin{cases} NNI(y) \geq 2^{bh(x)-1} - 1 \\ NNI(z) \geq 2^{bh(x)-1} - 1 \end{cases}$$

E quindi la seguente equazione ha senso:

$$NNI(y) + NNI(z) \geq 2^{bh(x)-1} - 1 + 2^{bh(x)-1} - 1$$

Sommiamo ambo i membri 1:

$$1 + NNI(y) + NNI(z) \geq 1 + 2^{bh(x)-1} - 1 + 2^{bh(x)-1} - 1$$

e sapendo che il numero di nodi interni di un albero equivale a uno più la somma dei nodi interni dei suoi due sotto-alberi (ovvero $NNI(x) = 1 + NNI(y) + NNI(z)$) otteniamo:

$$NNI(x) \geq 2 * 2^{bh(x)-1} - 1$$

$$\iff NNI(x) \geq 2^{bh(x)} - 1$$

cvd.

Dimostrazione 2 L'albero Red-Black di altezza fissata $bh(x)$ con meno nodi possibile che posso costruire è l'albero pieno con tutti nodi neri $[h(x) = bh(x)]$. In un albero pieno, il numero di nodi interni è

$$2^{bh(x)+1} - 1$$

ma a questo punto per conoscere il numero di nodi interni ci basta calcolare il numero di nodi dell'albero di altezza $bh(x) - 1$

$$NNI(x) = 2^{(bh(x)-1)+1} - 1 = 2^{bh(x)} - 1$$

A questo punto possiamo affermare che qualunque albero RB di altezza nera fissata, $bh(x)$ avrà perlomeno $2^{bh(x)} - 1$ nodi interni.

1.2 Dimostrazione $bh(x) \geq \frac{h(x)}{2}$

Fissato $bh(x)$, l'altezza minima che è possibile avere è $bh(x) = h(x)$, ovvero quando il percorso da x a una foglia comprende solo nodi neri, e quindi banalmente

$$bh(x) \geq \frac{h(x)}{2}$$

L'altezza massima che è possibile avere è quando si alterna un nodo nero con uno rosso in quanto per una delle proprietà degli alberi Red-Black non è possibile avere due nodi rossi consecutivi. Avremo quindi che ci sarà un nodo rosso per ogni nodo nero:

$$2bh(x) = h(x) \iff bh(x) = \frac{h(x)}{2}$$

e quindi

$$bh(x) \geq \frac{h(x)}{2}$$

vale.

1.3 Dimostrazione $h \leq 2 * log_2(n+1) \in O(log_2(n))$

Sapendo che $bh(x) \geq \frac{h(x)}{2}$ e sapendo che la funzione esponenziale è una funzione monotona, possiamo scrivere

$$2^{bh(x)} \geq 2^{\frac{h(x)}{2}}$$

Sottraiamo 1 ambo i membri:

$$2^{bh(x)} - 1 \geq 2^{\frac{h(x)}{2}} - 1$$

Da questo e dalla dimostrazione 1.1 possiamo dire che

$$NNI(x) \geq 2^{bh(x)} - 1 \geq 2^{\frac{h(x)}{2}} - 1$$

e per transitività:

$$NNI(x) \geq 2^{\frac{h(x)}{2}} - 1$$

Impostiamo x essere la radice dell'albero. Il numero di nodi interni in un albero RB equivale al numero di dati effettivamente presenti (in quanto le foglie non contengono informazione). Quindi $NNI(X) = n$, dunque:

$$n \geq 2^{\frac{h}{2}} - 1$$
$$\iff n + 1 \geq 2^{\frac{h}{2}}$$
$$\iff log_2(n+1) \geq \frac{h}{2}$$
$$\iff h \leq 2log_2(n+1)$$
$$\iff h \in O(log_2(n))$$

cvd.

References

- "Introduction to Algorithms" by Cormen, Leiserson, Rivest, and Stein