

Figura 11  
Albero bilanciato per  $\Delta=2$ . Risulta ad esempio:  $s_A - d_A = 2 - 3 = -1$ ;  $s_B - d_B = 2 - 2 = 0$ ;  $s_C - d_C = 3 - 1 = 2$ .

to nel paragrafo 4.2.3 gli alberi binari perfettamente bilanciati  $\mathcal{B}_k$  e quasi perfettamente bilanciati  $\mathcal{Q}_k$ , mostrando che tali strutture esibiscono un legame logaritmico tra il livello massimo  $k$  delle foglie e il numero  $n$  di nodi:  $k \in O(\log_2 n)$ . [5.14]

Ci chiediamo ora in che misura si possa alterare il perfetto bilanciamento dell'albero, senza compromettere la validità della [5.14]. Gli alberi che soddisfano tale legame presentano ovvio interesse tra gli alberi di decisione e, come vedremo in seguito, si rivelano di estrema importanza nei problemi di ricerca di dati.

Poniamo dunque la seguente definizione. Un albero è *bilanciato* in ragione di una prefissata costante  $\Delta \geq 0$  se per ogni suo nodo  $A$ , detti  $s_A$  e  $d_A$  i massimi livelli che si incontrano scendendo da  $A$  per il ramo sinistro e per il ramo destro, si ha  $|s_A - d_A| \leq \Delta$ .<sup>3</sup>

Per esempio è bilanciato per  $\Delta=2$  l'albero di figura 11. In particolare, gli alberi  $\mathcal{Q}_k$  sono bilanciati per ogni  $\Delta \geq 0$  e gli alberi  $\mathcal{B}_k$  per ogni  $\Delta \geq 1$ . Per controllare come cresce il livello massimo  $k$  in funzione del numero  $n$  di nodi nel caso peggiore, dobbiamo individuare, per ogni valore di  $\Delta$ , una famiglia di alberi bilanciati  $\mathcal{F}_k^\Delta$  che raggiungano ciascun livello  $k$  con minimo valore di  $n$ .

<sup>3</sup> Gli alberi bilanciati per  $\Delta=1$  sono meglio noti come alberi AVL, dai nomi dei matematici sovietici Adelson-Velskij e Landis (1962) che per primi ne proposero l'impiego.

Per  $\Delta=0$  gli  $\mathcal{F}_k^0$  coincidono banalmente con i  $\mathcal{B}_k$ . Per  $\Delta=1$  gli  $\mathcal{F}_k^1$  hanno la forma:

$\mathcal{F}_0^1 = \circ \quad k=0, n=1$

$\mathcal{F}_1^1 = \circ - \circ \quad k=1, n=2$

$\mathcal{F}_2^1 = \circ - \circ - \circ \quad k=2, n=4$

$\mathcal{F}_3^1 = \circ - \circ - \circ - \circ \quad k=3, n=7$

...

e possono essere in genere definiti nella forma ricorrente:

$\mathcal{F}_0^1 = \circ$

$\mathcal{F}_1^1 = \circ - \circ$

$\mathcal{F}_k^1 =$

$\mathcal{F}_{k-1}^1 - \mathcal{F}_{k-2}^1$

[5.15]

Gli  $\mathcal{F}_k^\Delta$  prendono il nome di *alberi di Fibonacci*, in virtù della legge di formazione [5.15] ove  $\mathcal{F}_k^\Delta$  è ottenuto per composizione di  $\mathcal{F}_{k-1}^\Delta$  e  $\mathcal{F}_{k-2}^\Delta$ . Ogni nodo  $A$  interno a  $\mathcal{F}_k^\Delta$  presenta il massimo "sbilanciamento" ammesso, cioè  $|s_A - d_A| = 1$  (per ogni foglia  $B$  è necessariamente  $|s_B - d_B| = 0$ ). Ciò garantisce che questi alberi sono costruiti con il minimo possibile numero di nodi, e raggiungono quindi il massimo livello  $k$  a pari  $n$ .

La [5.15] non è che un esempio di formazione delle famiglie  $\mathcal{F}_k^\Delta$ , e per  $\Delta \geq 1$  generico assume la forma:

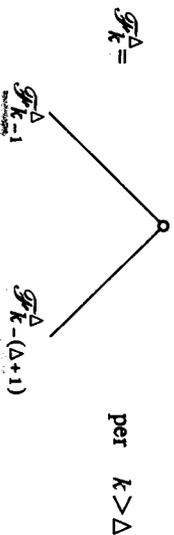
$$\mathcal{F}_0^\Delta = \circ \quad \text{per } k=0$$

$$\text{--- livello } 0$$

$$\mathcal{F}_k^\Delta = \text{--- livello } k \quad \text{per } 1 \leq k \leq \Delta$$

$$\text{--- livello } k$$

$$[5.16]$$



Anche per gli alberi  $\mathcal{F}_k^\Delta$  possiamo in genere osservare che ogni nodo  $A$  presenta il massimo sbilanciamento ammesso. Si ha infatti  $|s_A - d_A| = \Delta$  per ogni nodo interno, ad eccezione che per i nodi a distanza minore di  $\Delta$  dalle foglie, ove tale sbilanciamento è necessariamente minore. Gli  $\mathcal{F}_k^\Delta$  rappresentano quindi il caso pessimo da analizzare, poiché raggiungono il massimo livello a pari numero di nodi.

Detto  $n_k$  il numero di nodi di  $\mathcal{F}_k^\Delta$  per  $\Delta$  generico, dalle [5.16] ricaviamo immediatamente le relazioni

$$n_k = k + 1, \quad \text{per } 0 \leq k \leq \Delta, \quad [5.17]$$

$$n_k = n_{k-1} + n_{k-(\Delta+1)} + 1, \quad \text{per } k > \Delta. \quad [5.17']$$

Il termine  $+1$  che appare nel secondo membro della [5.17'] tiene conto del nodo radice di  $\mathcal{F}_k^\Delta$ . La relazione di ricorrenza [5.17'] è lineare non omogenea, di ordine  $\Delta + 1$ , e deve essere risolta rispettando le  $\Delta + 1$  condizioni iniziali [5.17].

Confrontando la [5.17'] con la forma generale [5.11] rileviamo nel caso presente:  $a_1 = 1, a_2 = \dots = a_{\Delta+1} = 0, a_{\Delta+1} = 1$ , e il termine noto  $h = 1$ . Poiché  $\sum_{i=1}^{\Delta+1} a_i = 2 \neq 1$ , si applica il metodo indicato nelle [5.12] e [5.13]. Calcolato  $h / (\sum_{i=1}^{\Delta+1} a_i - 1) = 1$ , si studia la successione  $y_k$  tale che

$$n_k = y_k - 1 \quad [5.18]$$

attraverso la relazione omogenea

$$y_k = (k+1) + 1 = k+2, \quad \text{per } 0 \leq k \leq \Delta, \quad [5.19]$$

$$y_k = y_{k-1} + y_{k-(\Delta+1)}, \quad \text{per } k > \Delta. \quad [5.19']$$

Imponendo una soluzione del tipo  $y_k = cz^k$ , la [5.19'] dà luogo all'equazione algebrica (vedi relazione [5.8])

$$z^{\Delta+1} - z^{\Delta} - 1 = 0 \quad [5.20]$$

che ha radici  $z_1, \dots, z_{\Delta+1}$ . La combinazione lineare

$$y_k = \sum_{i=1}^{\Delta+1} c_i z_i^k$$

fornisce la soluzione più generale della [5.19'], ove le costanti  $c_i$  si determinano imponendo le condizioni iniziali [5.19]. Il numero di nodi  $n_k$  si ricava ora dalle [5.18].

La crescita asintotica della soluzione si determina tuttavia più facilmente, poiché uno studio dell'equazione [5.20] mostra che, per qualsiasi valore  $\Delta \geq 1$ , essa ha una e una sola radice reale positiva  $z_1 > 1$ , mentre tutte le altre radici rispettano la relazione:  $|z_i| < z_1, 2 \leq i \leq \Delta + 1$  (vedi Luccio e Pagli, 1976).

Abbiamo dunque, per  $k \rightarrow \infty$ :

$$y_k \cong c_1 z_1^k$$

ovvero, combinando questo risultato con la [5.18]:

$$\begin{aligned} k &\cong (1/\log_2 z_1) (\log_2 (n_k + 1) - \log_2 c_1) \\ &\cong (1/\log_2 z_1) \log_2 n_k. \end{aligned} \quad [5.21]$$

In conclusione per tutti gli alberi  $\mathcal{F}_k^\Delta$ , e quindi *per tutti gli alberi bilanciati*

\* Per  $\Delta = 1$  la [5.19'] coincide con la relazione [5.11] di Fibonacci.

cizi (tra cui i precedenti raggiungono i massimi valori di  $k$ ), risulta

$$k \in O(\log_2 n_k).$$

In particolare, per  $1 \leq \Delta \leq 5$  abbiamo

$\Delta = 1$	2	3	4	5
$z_1 \cong 1,618$	1,466	1,380	1,325	1,285

da cui applicando la [5.21] si ricava

$k \cong 1,441 \log_2 n_k,$	per gli alberi	$\mathcal{P}_k^1;$
$k \cong 1,815 \log_2 n_k,$	per gli alberi	$\mathcal{P}_k^2;$
$k \cong 2,250 \log_2 n_k,$	per gli alberi	$\mathcal{P}_k^3;$
$k \cong 2,464 \log_2 n_k,$	per gli alberi	$\mathcal{P}_k^4;$
$k \cong 2,762 \log_2 n_k,$	per gli alberi	$\mathcal{P}_k^5.$

Notiamo infine che esempi semplicissimi di relazioni lineari non omogene di ordine uno sono le [3.3] e [3.4] incontrate nella determinazione del primo e secondo elemento di un insieme. In tali relazioni si ha però  $\sum_{i=1}^k a_i = 1$ , e il procedimento generale non può applicarsi nel modo visto.

Nel caso specifico delle [3.3] e [3.4] le espressioni lineari trovate per sostituzioni successive nel capitolo 3 sono le soluzioni generali delle ricorrenze; tali espressioni potrebbero ovviamente ricavarsi, a costo di maggior fatica, attraverso metodi generali di soluzione (vedi Reingold e altri, 1977).

## 5.2 Partizioni con lavoro costante di combinazione

Quando una relazione di ricorrenza di ordine costante esprime la complessità in tempo di una procedura ricorsiva, tale procedura non utilizza in genere una partizione bilanciata dei dati. E' questo il caso della relazione [3.4]: essa esprime il numero di passi eseguiti nel caso pessimo dalla procedura PRIMSEC2 (algoritmo 3.6) che determina il primo e secondo elemento di un insieme  $A$  ripartendo  $A$  in due sottoinsiemi di  $n-1$  e 1 elementi.

Abbiamo peraltro già discusso come il metodo di "divide et impera" sia efficiente proprio quando si basa su partizioni che, per essere bilanciate, danno in genere luogo a relazioni di ricorrenza di ordine non costante. In tali relazioni l'elemento generico  $x_n$  della successione è cioè definito

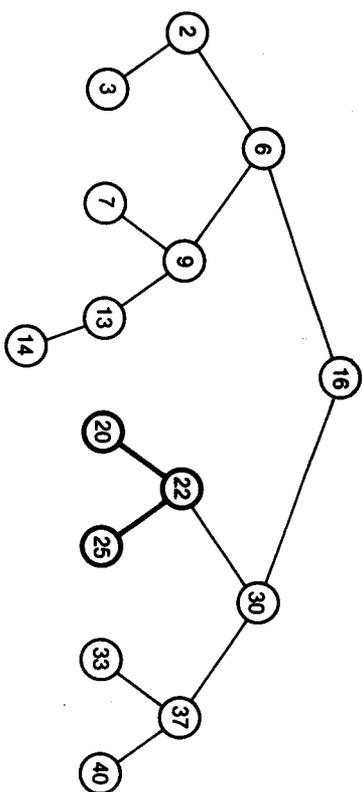


Figura 14

L'albero di figura 13 ribilanciato dopo l'inserzione dell'elemento 25 (la parte modificata è indicata con tratto spesso).

ta l'ordinamento degli elementi nei nodi che lo qualifica come albero binario di ricerca.

Esponiamo dunque il metodo generale di ribilanciamento per  $\Delta = 1$ . Si parte per ipotesi da un albero bilanciato. Nel percorso radice-foglia seguito per l'inserzione di un elemento  $z$  si individua il nodo critico, cioè il nodo a livello massimo che viola il bilanciamento dopo l'inserzione di  $z$  (nella fig. 13 il nodo 20 è critico dopo l'inserzione di 25). Per il nodo critico e i suoi sottoalberi si possono essenzialmente presentare le due situazioni indicate nella figura 15a e b, e le loro riflesses, ottenute scambiando i sottoalberi destri e sinistri.

Si eseguono ora le operazioni di ribilanciamento indicate nella figura 15. Il lettore potrà facilmente verificare la validità di queste operazioni controllando che, dopo il ribilanciamento:

1. l'albero è ancora un albero binario di ricerca, poiché lo spostamento dei nodi  $r$ ,  $s$  e dei sottoalberi  $\rho$ ,  $\sigma$ ,  $\tau$  (caso (a)), o dei nodi  $r$ ,  $s$ ,  $t$  e dei sottoalberi  $\rho$ ,  $\sigma$ ,  $\tau$ ,  $\omega$  (caso (b)), rispetta l'ordinamento degli elementi allocati nell'albero;
2. nella sezione dell'albero che è stata alterata (indicata in fig. 15) nessun nodo presenta sbilanciamento;
3. nella sezione dell'albero che è rimasta immutata (sezione non indicata in fig. 15) nessun nodo può presentare sbilanciamento, poiché nella sezione

ne alterata si è ripristinato il livello massimo  $k+1$  che era raggiunto nell'albero originale privo di  $z$ .

La ristrutturazione dei nodi 20, 22 e 25 nell'albero di figura 14 è un esempio di ribilanciamento del tipo indicato in figura 15a, con  $r=20$ ,  $s=22$ ,  $z=25$  e  $\rho$ ,  $\sigma$ ,  $\tau$  vuoti.

Non riportiamo esplicitamente l'algoritmo di ribilanciamento perché è piuttosto complesso (per approfondimenti vedi Knuth, 1973, § 6.2). È importante però notare che l'algoritmo è basato sulle trasformazioni di figura 15 che richiedono un tempo costante, poiché consistono nella variazione di tre rami dell'albero nel caso (a) (i rami che raggiungono  $r$ ,  $s$  e la radice del sottoalbero  $\sigma$ ) e di cinque rami nel caso (b) (i rami che raggiungono  $r$ ,  $s$ ,  $t$  e la radice dei sottoalberi  $\sigma$  e  $\tau$ ).

Poiché si può ribilanciare un albero binario di ricerca in tempo costante dopo l'inserzione di un nuovo elemento, questa struttura permette di eseguire una qualsiasi sequenza di operazioni di inserzione e ricerca in tempo di  $O(\log n)$  per ciascuna di esse.<sup>5</sup> In particolare il numero massimo di confronti tra elementi richiesto da tali operazioni coincide con il massimo livello  $k$  raggiunto dall'albero di Fibonacci  $\mathcal{F}_k$ , che rappresenta il caso pessimo per il problema. Così per  $\Delta=1$  il limite superiore al numero di confronti è circa  $1,44 \log_2 n$  (vedi i valori numerici riportati nel § 5.1.2 per la relazione [5.21]).

Accenniamo infine che è anche possibile eseguire eliminazioni di elementi in tempo logaritmico, ribilanciando l'albero secondo una procedura piuttosto complessa, per cui rimandiamo ancora a Knuth (1973).

Questi risultati fanno dell'albero binario di ricerca una delle strutture più importanti per la gestione di dati ordinati.

### 5.2.2 La moltiplicazione. Gli algoritmi di moltiplicazione hanno sem-

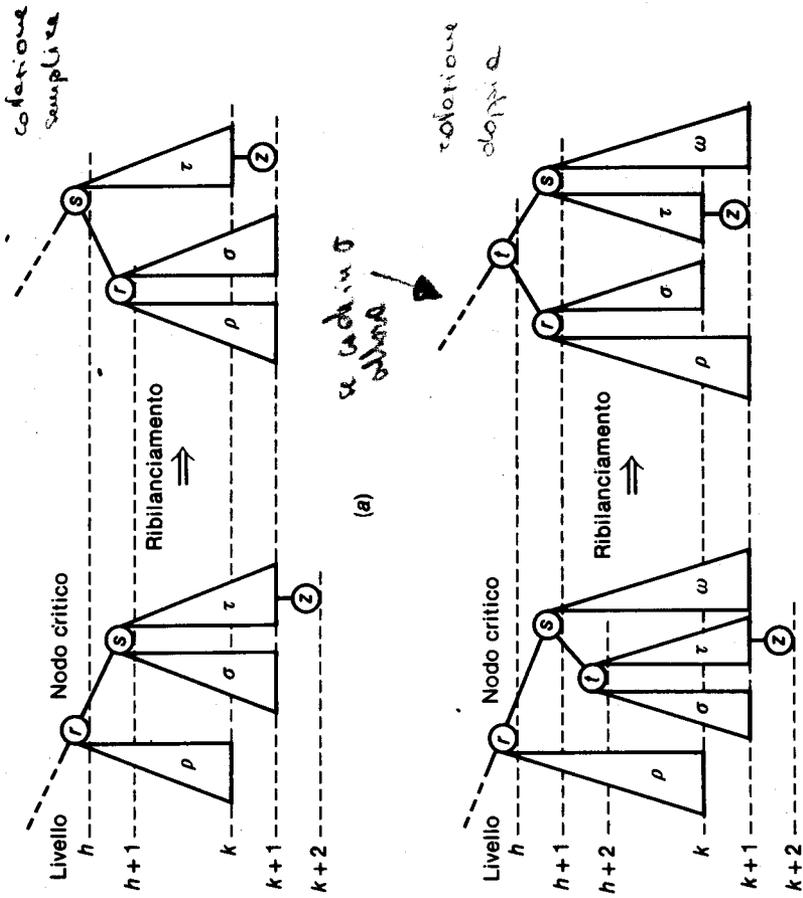


Figura 15

Le due situazioni di sbilanciamento per  $\Delta=1$ , e le operazioni di ribilanciamento:  $z$  è il nuovo elemento;  $r$  è il nodo critico;  $\rho$  e simili indicano interi sottoalberi. Nella situazione (b),  $z$  può essere inserito indifferentemente nel sottoalbero  $\tau$  o  $\sigma$ .