

15.1 Alberi binari

Le strutture dati che abbiamo fin qui esaminato sono lineari: ogni elemento della sequenza ha un successore e un predecessore, fatti salvi il primo e l'ultimo. Possiamo però pensare a qualcosa di più generale, in cui la relazione tra gli elementi della struttura non sia lineare.

Nel seguito utilizzeremo il termine *nodo* con il significato di elemento e il termine *arco* per indicare una connessione tra due nodi; con *etichetta* faremo invece riferimento al valore rappresentativo di ogni nodo.

Una delle strutture dati più note è l'*albero binario*, definito come un insieme B di nodi con le seguenti proprietà:

- B è vuoto oppure un nodo di B è scelto come radice;
- i rimanenti nodi possono essere ripartiti in due insiemi disgiunti B_1 e B_2 , essi stessi definiti come alberi binari.

B_1 e B_2 sono detti *sottoalberi* della radice. È importante sottolineare che il sottoalbero sinistro B_1 è distinto dal sottoalbero destro B_2 e questa distinzione permane anche se uno dei due è vuoto. La ricorsività della definizione stessa è evidente.

Un esempio di albero binario è presentato in Figura 15.1. Il nodo designato come radice ha etichetta 104, da esso si diparte il sottoalbero sinistro con radice 32 e il sottoalbero destro con radice 121. Si dice che i nodi con etichette 32 e 121 sono *fratelli* e sono rispettivamente il figlio sinistro e il figlio destro del nodo con etichetta 104. L'albero che ha come radice 32 ha ancora due sottoalberi con radici 23 e 44, i quali non hanno figli o, in altre parole, hanno come figli alberi vuoti. L'albero con radice 121 non ha figlio sinistro e ha come figlio destro il sottoalbero con radice 200, il quale a sua volta ha come figlio sinistro il sottoalbero con radice 152 e non ha figlio destro. Il sottoalbero che ha come radice 152 non ha figli.

I nodi da cui non si dipartono altri sottoalberi (non vuoti) sono detti *nodi terminali* o *foglie*. Nell'esempio sono quelli etichettati con: 23, 44 e 152.

Si chiamano *visite* le scansioni dell'albero che portano a percorrerne i vari nodi. L'ordine in cui questo avviene distingue differenti tipi di visite.

La visita *in ordine anticipato* di un albero binario viene effettuata con il seguente algoritmo:

```
anticipato( radice dell'albero )
  Se l'albero non è vuoto:
    Visita la radice
    anticipato( radice del sottoalbero sinistro )
    anticipato( radice del sottoalbero destro )
```

in cui abbiamo evidenziato la ricorsività della visita. Nel caso dell'albero di Figura 15.1 la visita in ordine anticipato dà la sequenza: 104, 32, 23, 44, 121, 200, 152.

La visita *in ordine differito* invece è così descritta:

```
differito( radice dell'albero )
  Se l'albero non è vuoto:
    differito( radice del sottoalbero sinistro )
    differito( radice del sottoalbero destro )
  Visita la radice
```

Applicando il procedimento sull'albero di esempio abbiamo: 23, 44, 32, 152, 200, 121, 104.

Concludiamo con un terzo tipo di visita, *in ordine simmetrico*, il cui algoritmo è:

```
simmetrico( radice dell'albero )
  Se l'albero non è vuoto:
    simmetrico( radice del sottoalbero sinistro )
    Visita la radice
    simmetrico( radice del sottoalbero destro )
```

che restituisce, con riferimento all'esempio di Figura 15.1: 23, 32, 44, 104, 121, 152, 200.

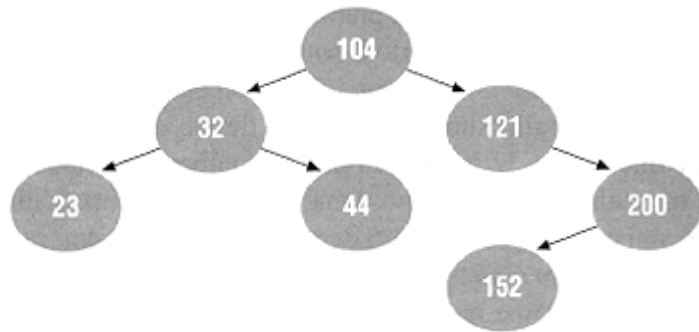


Figura 15.1 Esempio di albero binario