

Esercizi

3. Alberi binari, alberi di ricerca

Esercizio 3.1

Scrivete una funzione ricorsiva che, ricevendo il riferimento a un albero binario contenente numeri interi, restituisca la somma di tutti i valori pari contenuti nei nodi dell'albero.

Esercizio 3.2

Disegnate l'albero binario di ricerca ottenuto inserendo uno dopo l'altro in un albero inizialmente vuoto i numeri 15 7 2 16 13 1 4 20 17.

Indicate l'elenco dei valori contenuti nei nodi effettuando le visite in ordine anticipato, in ordine simmetrico e in ordine posticipato. Indicate l'elenco dei nodi incontrati effettuando la visita in ampiezza.

Esercizio 3.3

Ripetete l'esercizio 3.2 utilizzando i numeri 20 11 18 16 17 3 30 4 19 25. Disegnate poi l'albero che si ottiene rimuovendo, nell'ordine, i valori 20 18 dall'albero ottenuto.

Esercizio 3.4

Ripetete gli esercizi 3.2 e 3.3 effettuando gli inserimenti in modo da ottenere, a ciascun passo, un albero AVL.

Esercizio 3.5

Disegnate un albero di ricerca *perfettamente bilanciato* contenente tutti i numeri da 1 a 9.

Esercizio 3.6

Disegnate un albero di ricerca *perfettamente bilanciato* contenente i numeri 1 4 5 9 13 17 23 29 30.

Esercizio 3.7

Disegnate un albero di ricerca *perfettamente bilanciato* contenente i numeri 1 3 5 7 9 11 13 15 17 19 21 23 25.

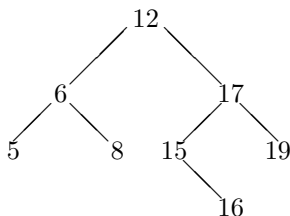
Esercizio 3.8

Disegnate l'albero AVL che si ottiene inserendo, a partire da un albero AVL inizialmente vuoto, tutti i numeri nell'ordine da 1 a 10.

Nota: non si richiede solo di ottenere un albero AVL contenente i numeri da 1 a 10, ma di inserire tali numeri in un albero AVL. Pertanto ad ogni inserimento, l'albero ottenuto dovrà essere un albero AVL (come per l'esercizio 3.4).

Esercizio 3.9

Considerate il seguente albero binario di ricerca:



Indicate l'elenco dei valori contenuti nei nodi effettuando le visite in ordine anticipato, in ordine simmetrico e in ordine posticipato. Indicate l'elenco dei nodi incontrati effettuando la visita in ampiezza.

Disegnate l'albero che si ottiene rimuovendo, nell'ordine, i valori 12 6 dall'albero.

Esercizio 3.10

Dimostrate che se in un albero binario ogni nodo interno possiede entrambi i figli, allora il numero delle foglie supera di 1 il numero di nodi interni. Che relazioni ci sono tra numero di nodi interni e numero di foglie nel caso ci possano essere nodi interni con un unico figlio?

Esercizio 3.11

Considerate un albero binario contenente n nodi, in cui i nodi contengono puntatori ai figli sinistro e destro. Esprimate il numero di puntatori null presenti nei nodi dell'albero in funzione di n .

Esercizio 3.12

Progettate un algoritmo *iterativo* per la visita in *ordine simmetrico* di tutti i nodi di un albero binario.

Suggerimento: come nella visita in ordine anticipato iterativa, presentata a lezione, è utile una struttura a pila, sulla quale si possono memorizzare le radici dei sottoalberi già raggiunte durante l'attraversamento, ma che devono essere ancora visitate.

Esercizio 3.13

Progettate un algoritmo *iterativo* per la visita in *ordine posticipato* di tutti i nodi di un albero binario.

Suggerimento: potete procedere in maniera simile all'esercizio precedente. Tuttavia, per ogni radice memorizzata sulla pila, è utile anche ricordare se sia stato già visitato il sottoalbero sinistro o no.

Esercizio 3.14

Disegnate un qualsiasi albero 2-3 contenente i valori 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31. Disegnate un albero 2-3 con gli stessi valori e il minimo numero possibile di nodi. Disegnate un albero 2-3 con gli stessi valori e il massimo numero di nodi.

Esercizio 3.15

Quant'è l'altezza minima di un albero 2-3 in cui sono memorizzate 20 chiavi? E l'altezza massima? Rispondete alle stesse domande nel caso di 200 e 2000 chiavi.

Esercizio 3.16

Ripetere gli esercizi 3.2 e 3.8 costruendo un albero 2-3.

Esercizio 3.17

Disegnate il B-albero di ordine (o grado minimo) $t = 2$ che si ottiene a partire dall'albero vuoto inserendo, nell'ordine, i valori 12 14 20 25 8 13 10 11 26 22 19 24 27 15 16 21.

Esercizio 3.18

Ripetete l'esercizio 3.17 con $t = 3$.

Esercizio 3.19

Calcolate il numero massimo di chiavi che possono essere collocate in un B-albero di altezza h e ordine t (potete ispirarvi al calcolo del numero minimo, presentato a lezione e nel Lemma 6.6 sul libro).