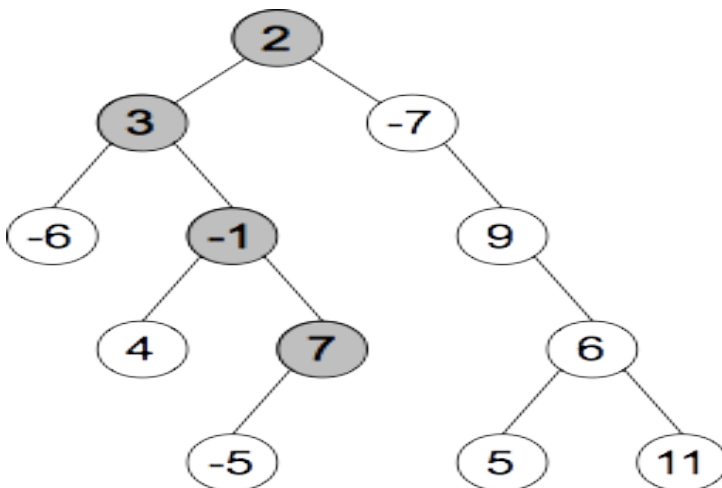


Esercizio1

Si consideri un albero binario non vuoto in cui a ciascun nodo v è associato un numero reale $v.val$. Scrivere un algoritmo che, dato in input l'albero T e un numero reale x , restituisce *true* se e solo se esiste un cammino dalla radice ad un qualsiasi nodo in cui la somma dei valori associati ai nodi del cammino sia esattamente uguale a x . Ad esempio, considerando l'albero T seguente e dato $x = 11$



l'algoritmo deve restituire *true*, in quanto il cammino composto dai nodi evidenziati ha somma x . Si può assumere che un cammino vuoto (cioè che non attraversa alcun nodo) abbia somma 0.

Esercizio 2

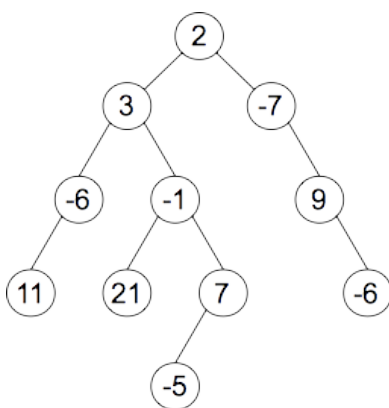
Si consideri un albero binario di ricerca, non bilanciato, contenente inizialmente i valori $\{1, 2, 3, 4, 5, 6, 7\}$.

1. Disegnare l'albero binario di ricerca di altezza minima contenente i valori dati;
2. Disegnare la struttura dell'albero dopo ciascuna delle seguenti operazioni (ogni operazione deve essere svolta sull'albero che si ottiene dopo quella precedente; la prima operazione va effettuata sull'albero di cui al punto 1).
 - a. Inserimento del valore 10
 - b. Cancellazione del valore 1
 - c. Inserimento del valore 9

- d. Cancellazione del valore 5
- e. Inserimento del valore 1

Esercizio 3

Si consideri un albero binario arbitrario in cui a ciascun nodo v sia associato un numero reale $v.val$. Scrivere un algoritmo ricorsivo che, dato in input l'albero T , restituisce il numero di nodi che contengono un valore negativo (cioè strettamente minore di zero). Ad esempio, considerando l'albero seguente:



l'algoritmo deve restituire 5, in quanto ci sono esattamente 5 nodi che contengono valori negativi. Nel caso di albero vuoto l'algoritmo deve restituire 0.

Esercizio 4

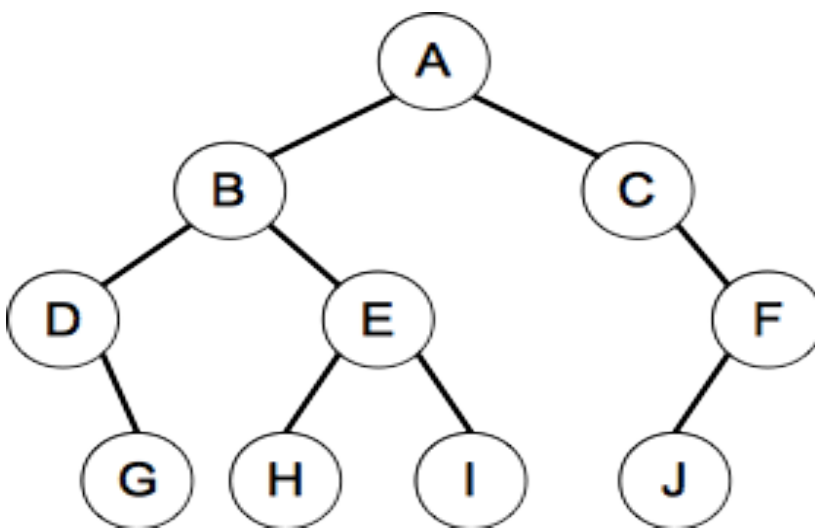
Si consideri un albero binario di ricerca non bilanciato, inizialmente vuoto.

Disegnare l'albero che risulta dopo l'inserimento delle chiavi seguenti (si assuma che ogni operazione di inserimento venga svolta sull'albero risultante al termine dell'inserimento precedente; la prima viene svolta sull'albero vuoto):

- . a) Inserimento 10
- . b) Inserimento 13
- . c) Inserimento 11
- . d) Inserimento 5
- . e) Inserimento 6
- . f) Inserimento 7
- . g) Inserimento 1

Esercizio 5

Si consideri il seguente albero binario:



Scrivere i nomi dei nodi come comparirebbero durante una visita in profondità in ordine anticipato (pre-visita: visita radice; visita ricorsiva sottoalbero sinistro; visita ricorsiva sottoalbero destro)

Scrivere i nomi dei nodi come comparirebbero durante una visita in profondità in ordine posticipato (post-visita: visita ricorsiva sottoalbero sinistro; visita ricorsiva sottoalbero destro; visita radice)

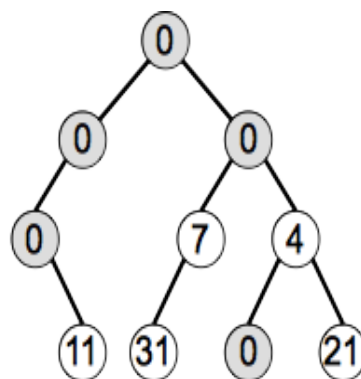
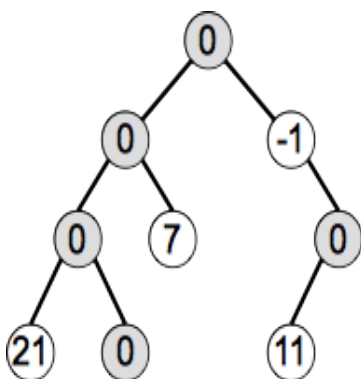
Esercizio 6

Si consideri un albero binario arbitrario, non vuoto, in cui ad ogni nodo t è associato un valore intero $t.val$. L'albero non è necessariamente un albero binario di ricerca, e potrebbe non essere bilanciato. Possono essere presenti più nodi con gli stessi valori. L'albero è rappresentato mediante puntatori: ogni nodo contiene un intero e i puntatori ai figli, che possono valere NULL.

Scrivere un algoritmo efficiente che, dato in input un puntatore alla radice dell'albero, restituisce *true* se e solo se

esiste un cammino dalla radice a una foglia composto interamente da nodi contenenti il valore zero.

Ad esempio, nel caso dell'albero a sinistra l'algoritmo restituisce *true*, mentre nel caso dell'albero a destra l'algoritmo restituisce *false* (per comodità i nodi contenenti zero sono evidenziati).



Esercizio 7

Si consideri un albero binario T in cui ad ogni nodo v è associato un valore intero $v.val$.

Scrivere un algoritmo ricorsivo che, dato in input T e un intero r , restituisce il numero di nodi che hanno valore esattamente uguale a r .

(Attenzione: T non è un albero binario di ricerca, quindi i nodi possono contenere valori qualsiasi senza un criterio particolare.)

Determinare il costo asintotico dell'algoritmo proposto, motivando la risposta

ESERCIZIO 8

Si consideri un albero binario, non vuoto, in cui a ciascun nodo i sia associato un

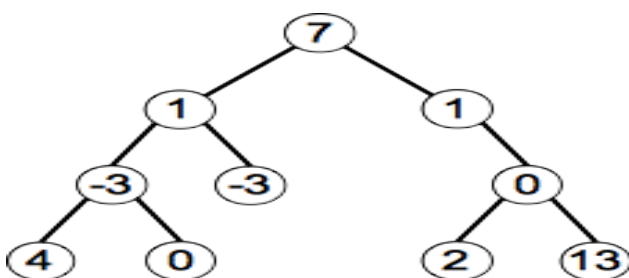
numero reale *i.real*. Scrivere un algoritmo ricorsivo che, dato in input l'albero T , restituisce il massimo valore contenuto nelle foglie e il numero di volte che tale valore è presente nell'albero.

ESERCIZIO 9

Consideriamo un albero binario di struttura arbitraria T , in cui a ciascun nodo v sia associato un valore intero $v.val$.

L'albero è rappresentato mediante nodi con puntatori ai figli.

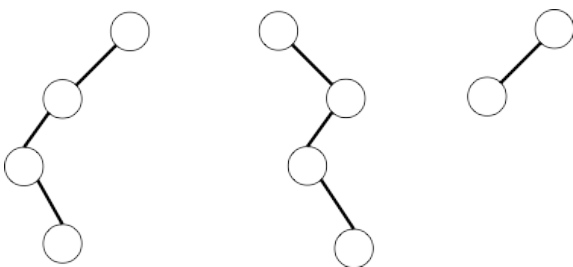
Scrivere un algoritmo ricorsivo che, dato in input un riferimento alla radice di T , restituisce il numero di nodi i cui figli hanno lo stesso valore.



l'algoritmo restituisce 2, in quanto ci sono due nodi (la radice, e il suo figlio sinistro) in cui entrambi i figli hanno lo stesso valore. [punti 4]

Determinare il costo asintotico dell'algoritmo proposto, motivando la risposta. [punti 2]

Un albero binario non vuoto si definisce *degenere* se ogni nodo diverso da una foglia ha un solo figlio (per convenzione, si assume che l'albero vuoto sia degenere). Ad esempio, gli alberi seguenti sono degeneri:



Scrivere un algoritmo ricorsivo che, dato in input l'albero T , restituisce true se l'albero è degenere, false altrimenti.

