Esercizi

Dato un albero binario, progettare un algoritmo efficiente che cancelli il figlio sinistro di ogni nodo se è una foglia e contiene la stessa chiave del nodo padre.

Dato un albero binario T, i cui nodi sono colorati di *bianco, rosso* o *verde,* progettare un algoritmo efficiente che stabilisca se esiste un cammino di tre nodi in T i cui colori formano la bandiera italiana. (Il cammino può partire da un nodo qualsiasi, non necessariamente dalla radice.)

Dato un albero binario i cui nodi sono colorati di *rosso* o di *nero,* progettare un algoritmo efficiente che calcoli il numero di nodi aventi lo stesso numero di discendenti rossi e neri. (Un nodo è discendente di se stesso.)

Un nodo v in un albero binario si dice **0-bilanciato** se le altezze dei sottoalberi radicati nei suoi due figli sono uguali. Dato un albero binario, progettare un algoritmo efficiente che determini il numero di nodi 0-bilanciati e analizzarne la complessità.

Progettare un algoritmo efficiente che stabilisca se un albero binario è un albero di Fibonacci, e analizzarne la complessità.
Suggerimento: *un albero binario è un albero di Fibonacci se e solo se tutti i nodi interni hanno fattore di bilanciamento uguale a 1 (in valore assoluto).*

Dato un array *a* di *n* elementi, progettare un algoritmo che costruisca ricorsivamente in tempo *O(n)* un albero binario bilanciato tale che *a[i]* sia l'(*i +1*)-esimo campo *u.dato* in ordine di visita anticipata. Considerare anche gli algoritmi per le altre visite: simmetrica, posticipata e per ampiezza.

Dato un albero binario, i cui nodi contengono chiavi intere, progettare un algoritmo che stabilisca se le chiavi nei nodi soddisfano la proprietà di heap di massimo, e analizzarne la complessità.

Un albero binario si dice **t-bilanciato** se per ogni suo nodo vale la proprietà: le altezze dei sottoalberi radicati nei suoi due figli differiscono per al più **t** unità.
Dato un albero binario, i cui nodi contengono solo i campi dato, sx e dx, progettare un algoritmo efficiente per determinare il minimo valore di **t** per cui l'albero risulti **t-bilanciato.**
