

Esercizi

su Alberi Binari, Alberi Binari di Ricerca (ABR), Alberi Bilanciati (AVL)

ASD 2007-2008

1. Progettare un algoritmo efficiente per stabilire se un albero binario é **completo a sinistra**.
2. Dato un albero binario, progettare un algoritmo efficiente che cancelli il figlio sinistro di ogni nodo se é una foglia e contiene la stessa chiave del nodo padre.
3. 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. (Nota: un nodo é discendente di se stesso).
4. 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.
5. Progettare un algoritmo efficiente che stabilisca se un albero binario é un ABR.
6. Progettare un algoritmo efficiente che stabilisca il successore di un nodo u in un ABR.
7. Dato un intervallo $[a, b]$ e un ABR, progettare un algoritmo efficiente per stampare in ordine crescente le chiavi dei nodi dell'albero che appartengono ad $[a, b]$.
8. Siano x e y con $x < y$, due chiavi in un ABR con radice u . Trovare la distanza (= numero di archi) tra il nodo di chiave x e il nodo di chiave y .
9. Progettare un algoritmo che inverta un ABR (ogni figlio dx diventa figlio sx) con chiavi tutte distinte.
10. Dato un ABR T , progettare un algoritmo ricorsivo efficiente che stabilisca se T é un AVL.