

Algoritmi e Strutture Dati
&
Laboratorio di Algoritmi e Programmazione

— Appello del 2 Settembre 2009 —

Esercizio 1 - ASD

1. Sia $T(n) = 3T(n/2) + 2n^2 + 3n \lg n$. Considerare ciascuna delle seguenti affermazioni e dire se è corretta o no. Giustificare la risposta.

- (a) $T(n) = \Omega(n)$
- (b) $T(n) = O(n^3)$
- (c) $T(n) = O(n \lg n)$

2. Sia $T(n) = 2T(n/3) + T(n/2) + n$. Verificare usando il metodo di sostituzione la correttezza della seguente affermazione. $T(n) = \Omega(n \lg n)$

Esercizio 2 - ASD

Discutere la correttezza di ciascuna delle seguenti affermazioni. Dimostrare formalmente la validità delle risposte date.

- 1. L'algoritmo di ricerca di una chiave in un albero R/B con n nodi ha complessità $\Theta(\lg n)$.
- 2. L'algoritmo di ricerca di una chiave in un albero BST con n nodi ha complessità $\Theta(\lg n)$.
- 3. L'algoritmo di ricerca di una chiave in un max-heap binario con n nodi ha complessità $\Theta(\lg n)$.
- 4. Esiste un algoritmo lineare che trasforma un albero BST in un albero binario completo che soddisfa la condizione di max-heap.

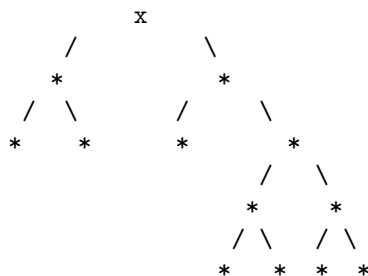
Esercizio 3 - ASD

Si sviluppi un algoritmo che verifica se un dato albero generale soddisfa la seguente proprietà:
per ogni nodo x si ha che x non ha figli oppure la chiave associata ad x è minore della chiave associata al più giovane figlio di x (cioè all'ultimo nella lista dei suoi figli).

2. Si dimostri la correttezza dell'algoritmo proposto.

Esercizio 4 - ASD

- 1) Si realizzi una algoritmo che dato un nodo x di un albero binario, calcola il numero di sottoalberi binari completi (non vuoti) dell'albero radicato in x .
 - 2) Si dimostri la correttezza dell'algoritmo.
- Esempio: Nell'albero seguente contiamo 11 sottoalberi completi.



Esercizio 1 (Laboratorio)

Scrivere una classe *QueueList* che realizza una coda con una lista concatenata. Scegliere il tipo di lista in modo che le operazioni *isEmpty*, *enqueue*, *dequeue*, *front* siano realizzate in tempo costante. Si richiede di:

- scrivere una classe *NodoLista* che memorizza il singolo nodo della lista concatenata;
- implementare la classe *QueueList* con un opportuno costruttore e le quattro operazioni indicate.

Esercizio 2 (Laboratorio)

Si consideri una tabella hash $T[0\dots 10]$ gestita con la funzione hash $\mathbf{h}(\mathbf{k}) = (\mathbf{k} \bmod 11)$ e in cui le collisioni sono gestite col metodo dell'indirizzamento aperto mediante *scansione quadratica*. Si richiede di:

1. scrivere la legge di scansione quadratica per la tabella data;
2. sapendo che la tabella è inizialmente vuota e che le costanti relative alla scansione quadratica hanno valori $c_1 = 3$ e $c_2 = 5$, inserire le seguenti chiavi:
 - $k_1 = 84$
 - $k_2 = 58$
 - $k_3 = 12$
 - $k_4 = 172$
 - $k_5 = 111$
 - $k_6 = 49$
 - $k_7 = 106$

Per ogni chiave specificare la posizione di inserimento e quanti tentativi di inserimento sono stati effettuati;

3. disegnare la tabella risultante e dire qual è il suo fattore di carico.