

APPELLO DEL 23 GIUGNO 2009

CHI È ESONERATO DEVE SVOLGERE SOLO GLI ESERCIZI DELLA PARTE 2 (TEMPO CONCESSO: 1H 30 MIN).
GLI ALTRI STUDENTI POSSONO SVOLGERE SOLO GLI ESERCIZI DELLA PARTE 1 (TEMPO CONCESSO: 1H 30 MIN),
COMPLETANDO L'ESAME IN LUGLIO, OPPURE L'ESAME COMPLETO (TEMPO CONCESSO: 3H).

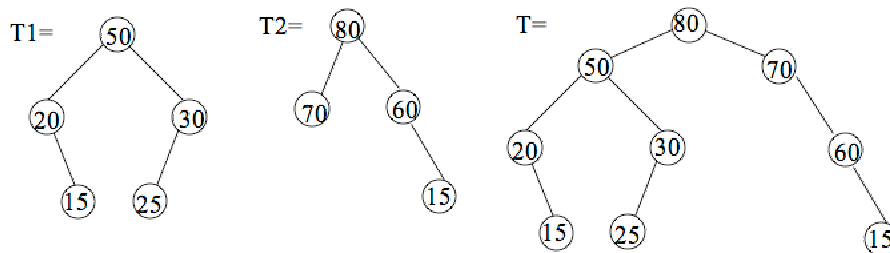
Parte 1, esercizio 1 (max 5 punti). Dimostrare per sostituzione che la seguente relazione di ricorrenza ha soluzione $T(n) = O(n \log n)$:

$$T(n) = \begin{cases} 1 & \text{se } n \leq 2 \\ T(n-1) + \log n & \text{altrimenti} \end{cases}$$

Parte 1, esercizio 2 (max 25 punti). Siano T_1 e T_2 due alberi binari che soddisfano la proprietà di ordinamento a heap (ogni nodo ha una chiave maggiore o uguale di quella dei suoi figli). Siano h_1 l'altezza di T_1 , h_2 l'altezza di T_2 , n_1 il numero di nodi di T_1 , e n_2 il numero di nodi di T_2 .

(A) (**max 10 punti**) Assumendo che gli alberi siano *rappresentati in memoria tramite record e puntatori ai figli sinistro e destro*, scrivere un algoritmo che combini T_1 e T_2 in un nuovo albero binario T che soddisfa ancora la proprietà di ordinamento a heap.

Ad esempio, dati gli alberi T_1 e T_2 , l'output T potrebbe essere:



(B) (**max 5 punti**) Dimostrare la correttezza dell'algoritmo proposto.

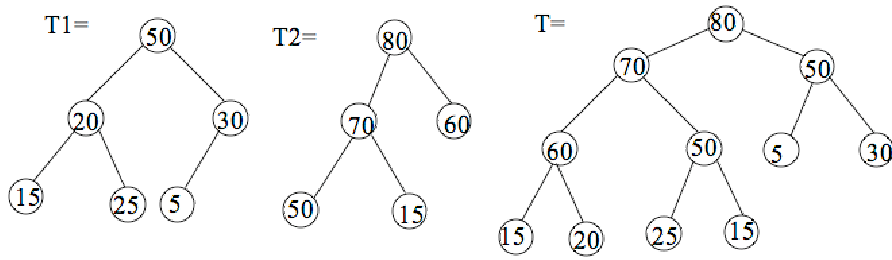
(C) (**max 5 punti**) Il tempo di esecuzione nel caso peggiore dell'algoritmo proposto è:

- $O(\max\{h_1, h_2\})$?
- $O(h_1 + h_2)$?
- $\Theta(n_1 + n_2)$?

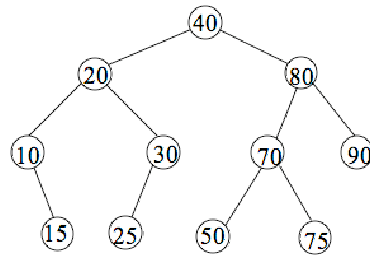
Motivare le risposte.

(D) (**max 5 punti**) Assumere ora che gli alberi siano due heap con proprietà di struttura forte memorizzati tramite due vettori posizionali H_1 e H_2 . Descrivere un algoritmo per combinare H_1 e H_2 in un unico heap H , utilizzando come sottoprocedure le funzioni sugli heap introdotte a lezione (`buildHeap`, `fixHeap`, `insert`, etc.). Qual è il tempo di esecuzione dell'algoritmo proposto?

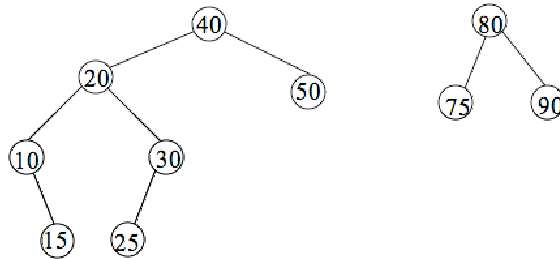
Ad esempio, dati gli heap T_1 e T_2 , l'output T potrebbe essere:



Parte 2, esercizio 1 (max 13 punti). Sia T un albero binario di ricerca di altezza h senza duplicati, e sia k una chiave. Descrivere ed analizzare un algoritmo `split(T,k)` che calcoli da T e k due alberi binari di ricerca T_1 e T_2 contenenti il primo le chiavi minori di k e il secondo le maggiori. La chiave k va eliminata. Ad esempio, se l'albero dato è quello disegnato sotto e $k = 70$:

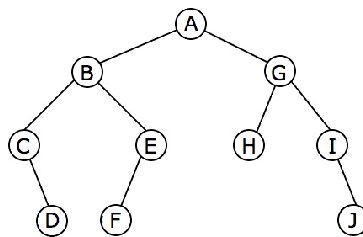


si possono ottenere i due alberi:



[Suggerimento: facendo risalire k alla radice tramite le rotazioni introdotte per gli alberi AVL si può ottenere un algoritmo con tempo di esecuzione $O(h)$.]

Parte 2, esercizio 2 (max 17 punti). Una visita di un albero binario è detta *in triplo ordine* se ogni nodo viene visitato in tre diversi momenti: prima del suo sottoalbero sinistro, dopo il sottoalbero sinistro ma prima del sottoalbero destro, dopo il sottoalbero destro. Ad esempio, la visita in triplo ordine dell'albero mostrato in figura



incontrerebbe i nodi nel seguente ordine:

A B C C D D D C B E F F F E E B A G H H H G I I J J J I G A

Sia X l'array di dimensione $3n$ (n è il numero di nodi dell'albero) prodotto dalla visita in triplo ordine. Sia inoltre M l'array di dimensione $3n$ tale che:

- $M[p] = 1$ se p indicizza la prima occorrenza in X del nodo $X[p]$;
- $M[p] = 2$ se p indicizza la seconda occorrenza in X del nodo $X[p]$;
- $M[p] = 3$ se p indicizza la terza occorrenza in X del nodo $X[p]$.

Nell'esempio precedente, gli array X e M sono:

$X = A B C C D D D C B E F F F E E B A G H H H G I I J J J I G A$
 $M = 1 1 1 2 1 2 3 3 2 1 1 2 3 2 3 3 2 1 1 2 3 2 1 2 1 2 3 3 3 3$

- (A) (**max 2 punti**) Descrivere ed analizzare un algoritmo che, dati in input n e l'array X , stampi tutte le foglie dell'albero rappresentato da X .
- (B) (**max 2 punti**) Si consideri l'array di dimensione n contenente la prima occorrenza in X di ciascun nodo dell'albero. Che cosa rappresenta tale array? E quello contenente la seconda occorrenza di ciascun nodo?
- (C) (**max 6 punti**) Si osservino le possibili coppie di valori in posizioni consecutive nell'array M . Per ciascuna coppia di valori (i, j) , con $1 \leq i \leq 3$ e $1 \leq j \leq 3$, si dica se è possibile che la coppia compaia in M oppure no. Motivare le risposte per tutte le nove coppie possibili.
- (D) (**max 7 punti**) Sia (i, j) una coppia di valori che possono comparire in posizioni consecutive nell'array M , e siano u e v i corrispondenti nodi. Si dica se nel passare da u a v durante la visita in triplo ordine si sta scendendo di un livello nell'albero, si sta rimanendo sullo stesso livello, o si sta risalendo di un livello. Ripetere il ragionamento per tutte le coppie possibili, motivando le risposte.