

Algoritmi e Strutture Dati (A.A. 2013-2014)

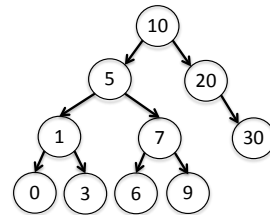
Corso di Laurea in Ingegneria dell'Informazione
Sapienza Università di Roma

Seconda prova di esonero (7/01/2014) – Durata 2 ore

Cognome: _____ Nome: _____ Matricola: _____	Autorizzo la pubblicazione del voto di questo esame sul sito web http://www.dis.uniroma1.it/~demetres/didattica/asd , secondo quanto prevede il decreto legislativo 196/2003 (codice in materia di protezione dei dati personali) che dichiaro di conoscere. In fede, _____
---	--

Domanda 1 [12 punti]

1. Fornire una definizione degli alberi di Fibonacci e dimostrare che un albero AVL con n nodi ha altezza $O(\log n)$.
2. Si consideri il seguente albero AVL T :
 - a. eliminare la chiave 10 da T
 - b. eliminare la chiave 20 da T
 - c. inserire la chiave 2 in T
 - d. inserire la chiave 8 in T



NB: le operazioni non vanno effettuate in sequenza, ma sempre a partire dall'albero iniziale T . Mostrare per ogni operazione l'albero risultante e il tipo di rotazioni eventualmente applicate, riportando per ogni nodo il fattore di bilanciamento prima e dopo l'operazione.

Domanda 2 [8 punti]

Durante un colloquio di lavoro per la possibile assunzione a Facebook, vi viene chiesto di descrivere come realizzereste un modulo software in grado di verificare se un utente ha accettato l'amicizia di un altro. Questa operazione va svolta molto frequentemente e deve pertanto essere supportata dal modulo software il più efficientemente possibile. Il modulo prende in input un grafo orientato che rappresenta la porzione della rete sociale di Facebook di interesse, dove ogni utente è rappresentato da un nodo e c'è un arco (u,v) se u ha accettato l'amicizia richiesta da v . Il grafo è dato in input sotto forma di lista di archi, ma può essere eventualmente convertita dal modulo software in un'altra rappresentazione che risulti più conveniente. Si assuma che la memoria RAM di lavoro disponibile al programma sia $S(n,m)=\Theta(m+n)$, dove n =numero di nodi e m =numero di archi. Tenendo presente che la rete è sparsa, cioè $m=O(n)$, come risolvereste il problema, assumendo in prima approssimazione che il grafo non cambia nel tempo? Discutere il tempo richiesto nel caso peggiore per ogni test che verifica se u ha accettato l'amicizia di v .

