Informatica Generale
Esame Scritto

docenti:
Giancarlo Bongiovanni, Tiziana Calamoneri, Ivano Salvo
Sapienza Università di Roma

23 giugno 2017

Esercizio 1 (10 punti) Si consideri la seguente equazione di ricorrenza:

$$T(n) = 8T(n/2) + n \text{ se } n > 1$$
$$T(1) = 2$$

e la si risolva utilizzando:

a. (**3 punti**) il metodo iterativo;
b. (**2 punti**) il metodo dell'albero;
c. (**1 punto**) il metodo principale;
d. (**2+2 punti**) il metodo di sostituzione.

Esercizio 2 (10 punti) Diremo che un albero binario di interi A è *minore per tracce* di un albero di interi B (notazione $A \preceq B$) se per ogni sequenza di etichette p in un cammino di A, esiste un cammino in B in cui incontro la stessa sequenza di etichette p.

Gli alberi A e B sono *equivalenti per tracce* se e solo se $A \preceq B$ e $B \preceq A$.

Scrivere una funzione C di prototipo: `int eqTrace(binTree A, binTree B);` che ritorna 1 se gli alberi in ingresso A e B sono equivalenti per tracce e 0 altrimenti.

Esempi: Presi gli alberi in Fig. 1, abbiamo chiaramente che $A \preceq C$ e $A \preceq B$. Osservate che la sequenza di etichette $\langle 11, 23, 19 \rangle$ si trova nell'albero B nel cammino contenente le etichette $\langle 11, 23, 19, 42 \rangle$. Questo cammino è il controesempio che mostra che $B \not\preceq A$ e $B \not\preceq C$. Infine anche $C \preceq A$ e quindi A e C sono equivalenti per tracce.

Suggerimento: Organizzare il codice in funzioni che risolvono sottoproblemi più semplici. Ovviamente è più semplice scrivere una funzione che implementa $\preceq$.

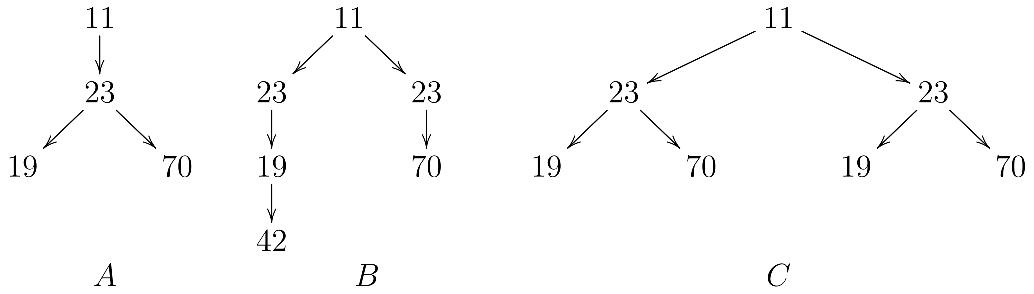

Figura 1: Alberi binari di esempio

Esercizio 3 (10 punti) Dati un grafo non orientato e connesso $G = (V, E)$, e due suoi nodi u e v, si definisce *distanza* tra u e v il numero di archi nel cammino più corto tra u e v in G. Si definisce inoltre *diametro* di G, denotato con $diam(G)$, come la massima distanza tra due nodi del grafo.

Sia dato un nodo s di G. Si assuma che una visita in ampiezza (BFS) di G a partire da s produce un albero ricoprente con $k + 1$ livelli (s è a livello 0 mentre la foglia più lontana da s è a livello k).

1. **(2 punti)** Dimostrare che $k \leq \text{diam}(G) \leq 2k$;

2. **(4 punti)** Progettare e descrivere (dettagliatamente a parole o in pseudocodice) un algoritmo che calcoli il diametro di un grafo;

3. **(3+1 punti)** Nel calcolare il costo computazionale dell'algoritmo proposto, discutere quale sia la struttura dati più adatta per memorizzare il grafo in questo contesto.