

FONDAMENTI DI PROGRAMMAZIONE - CdL in MATEMATICA

PROVA SCRITTA DEL 13/7/2009

Scrivere **in stampatello** COGNOME, NOME e MATRICOLA su ogni foglio consegnato

N.B.: Negli esercizi di programmazione, viene valutata anche la leggibilità del codice proposto. Inoltre, non è consentito l'uso di istruzioni che alterino il normale flusso dell'esecuzione all'interno di cicli che ne provochino l'uscita forzata).

Laddove è utilizzato, il tipo `boolean` è definito da `typedef enum {false, true} boolean;`

ESERCIZIO 1 (7 punti)

Si consideri la seguente grammatica G

$$G = (V = \{S, X\}, T = \{0\}, P = \{S \rightarrow XX, X \rightarrow XXX \mid 0\}, S)$$

- dire che tipo di grammatica è;
- mostrare, se esistono, due derivazioni per `aaaaa` e per `aaaa`;
- qual è il linguaggio generato da G ?
- mostrare, se esiste, una grammatica regolare che genera lo stesso linguaggio.

ESERCIZIO 2 (7 punti)

Si vuole scrivere una funzione **ricorsiva** che legge una sequenza di numeri interi terminata da 0 e produce in output il valore di verità `true` se il numero di elementi positivi nella sequenza è uguale al numero di elementi negativi della sequenza; produce in output il valore `false` altrimenti. Se ad esempio la sequenza letta è

16 9 -32 -2 4 -64

l'output atteso è `true`. Se invece la sequenza letta è

16 -3 12 -8 9 5 7

l'output atteso è `false`.

Si risolva il problema, tenendo presente che:

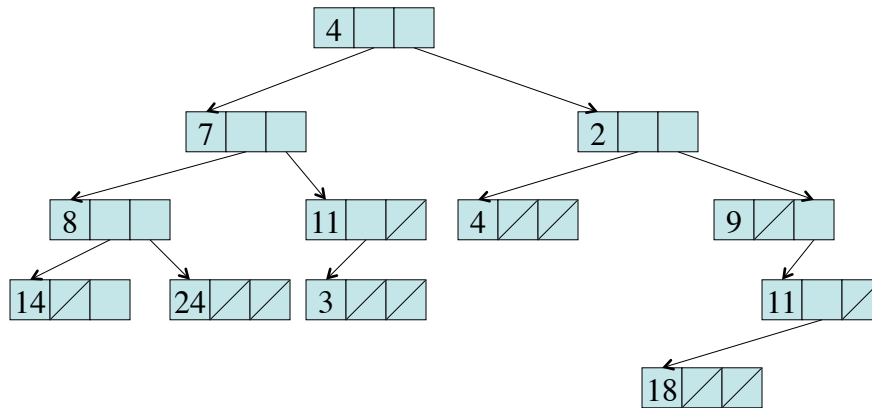
- non si possono utilizzare costrutti iterativi
- non si possono utilizzare variabili strutturate (ad es. vettori)
- non si possono utilizzare strutture dinamiche (ad es. liste)

ESERCIZIO 3 (7 punti)

Definire una funzione che, dato un array di interi e la sua lunghezza (che si suppone non nulla), verifica se il vettore contiene o meno tutti elementi distinti tra loro. Nella definizione di tale funzione l'unico costrutto iterativo utilizzabile è il costrutto **while**.

ESERCIZIO 4 (12 punti)

Si vuole rappresentare un **Albero Binario**. Ci serve quindi una struttura dinamica, in cui ogni elemento rappresenta un nodo, e quindi memorizza un numero intero, il puntatore al sottoalbero sinistro e il puntatore al sottoalbero destro. Un nodo foglia è caratterizzato dal fatto che non ha ulteriori sottoalberi. Di seguito riportiamo un esempio.



- (1 punti) Definire i tipi opportuni per la rappresentazione indicata.
- (5 punti) Scrivere una funzione **iterativa** che restituisca il puntatore corrispondente al nodo foglia più a sinistra dell'intero albero, o in mancanza di quello, il nodo interno più a sinistra. Nel caso dell'esempio il puntatore all'elemento con associato il numero 14. Se non esiste si deve restituire **NULL**.
- (6 punti) Scrivere una funzione **ricorsiva** che dato un albero binario e un intero x , restituisca il numero di volte in cui il valore x appare nei nodi dell'albero. Nel caso dell'esempio, se x fosse 11, il numero da restituire sarebbe 2. Se fosse invece 3, sarebbe 1.