

Modulo B (33 punti) Il tempo a disposizione è 1:15 h. Giustificare tutte le risposte, usando anche il retro dei fogli.

(15^{pts})

1. Si consideri un albero binario T . Definiamo un $PSD := Percorso Senza Diramazioni$ come un insieme di vertici v_1, v_2, \dots, v_m ($m \geq 1, v_i \in T$) dove ogni nodo è collegato al precedente e non ha più di un figlio, formalmente:

15 pts

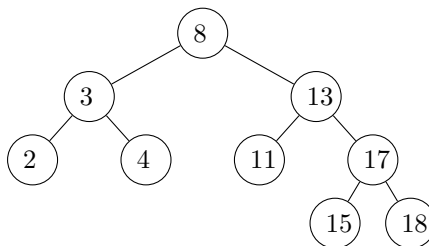
1. Per $2 \leq i \leq m$ il nodo v_i è figlio del nodo v_{i-1}
2. Per $1 \leq i \leq (m - 1)$ il nodo v_i ha un solo figlio
3. v_m non ha figli (quindi è una foglia)

- (a) (9 pts) Scrivere in pseudocodice una funzione ricorsiva `int MaxPSD(Tree T, <eventuali parametri aggiuntivi>)` che restituisca la lunghezza massima fra i PSD dell'albero binario T . Scrivere anche i valori degli eventuali parametri aggiuntivi per la prima chiamata.
- (b) (3 pts) Se T fosse un albero binario bilanciato (non necessariamente completo), quale sarebbe la sua (eventuale) lunghezza massima fra i PSD ?
- (c) (3 pts) Se l'albero binario T viene rappresentato con un array, preallocato per la dimensione massima dell'albero nel caso sia completo. Quanto spazio vuoto 'spreca' un PSD con m nodi?

(10^{pts})

2. Si consideri il seguente albero AVL:

10 pts



- (a) (5 pts) Cancellare l'elemento 8 e disegnate l'albero AVL risultante dalla cancellazione.
- (b) (5 pts) Inserite l'elemento 14 e disegnate l'albero AVL risultante dall'inserimento.

(8^{pts})

3. Dato il seguente albero binario in forma vettoriale $A = [0, 2, 6, 7, 3, 9, 5, 4, 1, 8]$:

8 pts

- (a) (5 pts) Costruire un $MaxHeap$ di A usando la funzione `build-heap` (la versione con i massimi) e scrivere i vettori restituiti dalle chiamate della `heapify`, senza contare le chiamate ricorsive.
- (b) (3 pts) Quanti confronti sono stati fatti nel punto precedente e in generale qual è l'ordine massimo di grandezza del numero di confronti per la costruzione di uno heap?