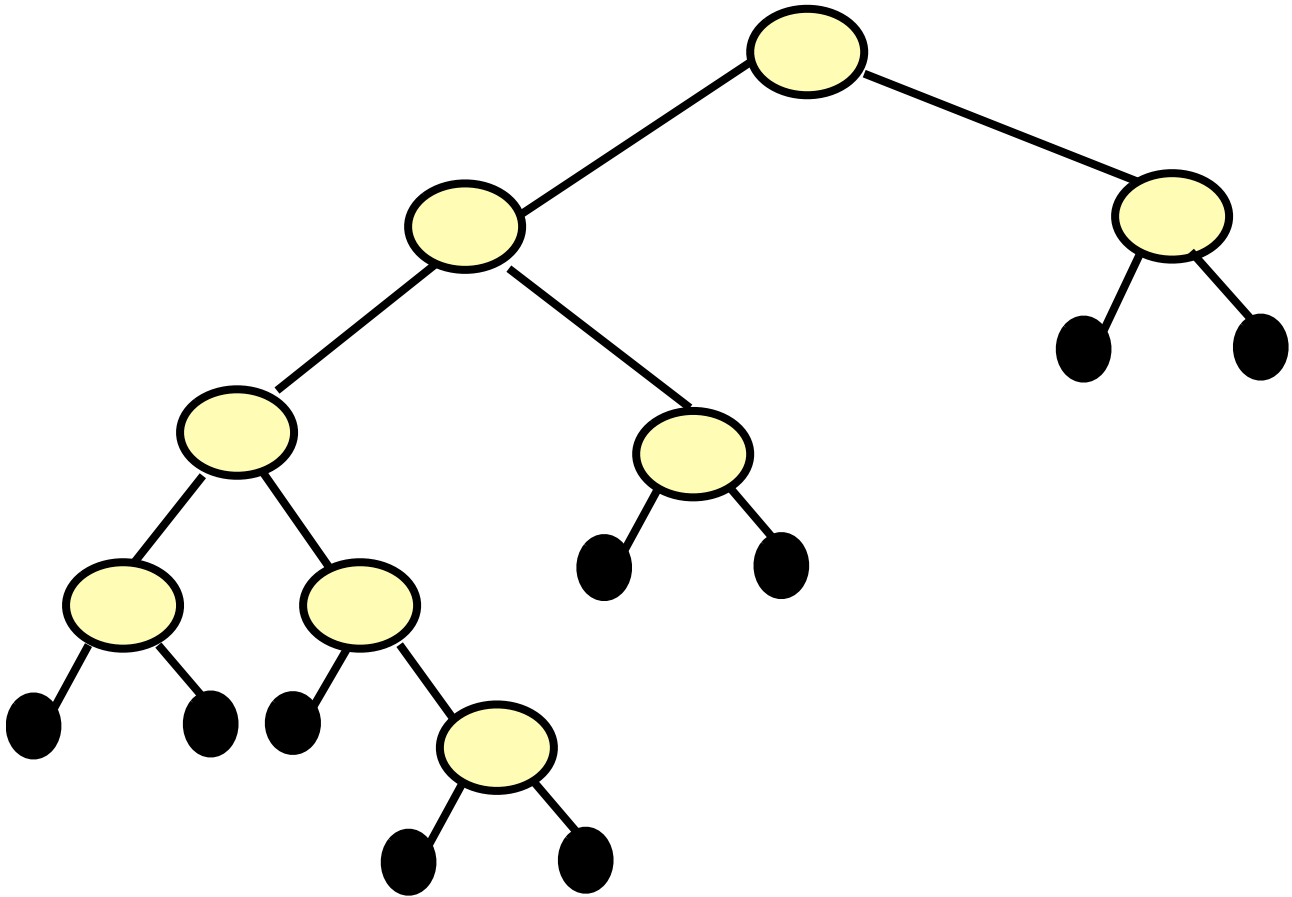


1. Alberi binari di ricerca e alberi rosso-neri

a. (punti 3) Si assegnino le chiavi 2, 3, 5, 7, 11, 13, 17, 19 ai nodi del sottostante albero T in modo che ne risulti un ABR



b. (punti 3) Si spieghi perchè T non può essere colorato in modo da ottenere un albero rosso-nero.

c. (punti 3) L'ABR T può essere trasformato in un albero rosso-nero con una singola rotazione. Si individui la rotazione dicendo su quale nodo fa perno e se si tratta di una rotazione sinistra o destra, si disegni l'albero ottenuto colorando ogni nodo di rosso o di nero.

2. Ottimalità degli algoritmi

(punti 5) Dato un vettore non ordinato A di n interi si scriva un algoritmo che costruisce un albero rosso-nero con gli elementi di A in $O(n \lg n)$.

Si dimostri che la complessità nel caso peggiore è in $\Omega(n \lg n)$.

Si può dire che l'algoritmo è ottimale? Si motivi la risposta.

(punti 4) Si determini un limite inferiore alla complessità di BUILD-MAX-HEAP, cioè della procedura di costruzione di un Max_Heap a partire da un array qualunque..

Si può dire che l'algoritmo è ottimale? Si motivi la risposta.