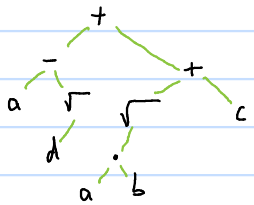
$f = a - \sqrt{d} + (\sqrt{a \cdot b} + c)$

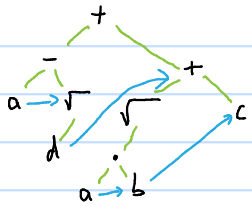
15 a)



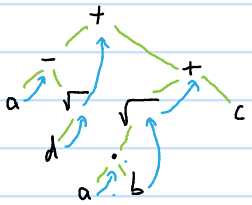

b) Preorder: $+ - a \sqrt{} d + \sqrt{} \cdot a b c$

Postorder: $a d \sqrt{} - a b \cdot \sqrt{} c + +$

c) Preorder Rechts-Fädelung




Inorder Rechts-Fädelung




16 a) Suchen der Einfügeposition:

Bei einem binären Suchbaum kann nach jedem Vergleich der Suchraum halbiert werden. Folglich ist die Laufzeit $O(\log n)$. Vorausgesetzt der Baum wird gleichmäßig aufgebaut und bekommt als Wurzel das Mittlerste Element des Zahlenbereichs der zu sortierenden Elemente und die Liste der Elemente ist gleichverteilt / unsortiert.

Das Erstellen des Baums mit $n$ Elementen ist somit in $O(n \cdot \log n)$

Die Ausgabe zur sortierten Liste ist $O(n)$. Gesamtaufwand: $O(n \log n + n) \Rightarrow O(n \log n)$

b) Suchen der Einfügeposition:

Im ungünstigsten Fall hat jeder Knoten des Baums nur ein Kind. Bei der Suche müssen alle Elemente bis zur Einfügeposition des Elements durchlaufen werden: Laufzeit einer verketteten Liste: $O(n)$

Beim Einfügen von $n$ bereits sortierten Elementen folgt daraus eine Laufzeit in $O(n^2)$

Zusätzlicher Ausgabedurchlauf (inorder) erhöht die Laufzeit wieder um $n$: $O(n^2 + n) \Rightarrow O(n^2)$