

AD [HA] zum 6. 11. 2013

Arne Struck, Lars Thoms

4. November 2013

1. a)

Es liegen k^l Blätter maximal in der l. Ebene. Von jedem Knoten gehen k Knoten ab, das führt zum folgenden: 0. Ebene (root): $1 = k^0$, 1. Ebene: $k = k^1$, 2. Ebene: $k \cdot k = k^2$, 3. Ebene: $k \cdot k \cdot k = k^3$... l.Ebene: k^l

b)

Der volle Baum hat $\sum_{i=0}^l k^i = \frac{k^{l+1}-1}{k-1} = \frac{k^l + k^l \cdot (k-1)}{k-1} = \frac{k^l-1}{k-1} + k^l$ Knoten, die Summe der Knoten aller Ebenen (eine volle Ebene bemisst sich, wie in a) dargestellt auf k^l).

c)

Der vollständige Baum hat $\sum_{i=0}^{l-1} k^i + c = \frac{k^l-1}{k-1} + k^l - k^l + c \mid c \in \mathbb{N} : 1 \leq c \leq k^l$ Blätter. Der vollständige Baum ist bis zu seiner vorletzten Ebene maximal gefüllt, deswegen die Summe bis $l-1$, c repräsentiert die Anzahl der Blätter in der letzten Ebene, welche zwischen einem (sonst wäre der Baum voll und hätte l-1 Ebenen) und k^l (ein voller Baum ist vollständig) Blättern.

d)

Der Baum hat n-1 Kanten, da jeder Knoten (bis auf den Wurzelknoten) eine Kante besitzt durch die er mit seinem Elternknoten verbunden ist.

2. a)

Die Laufzeit kann wie folgt (für OrderX) hergeleitet werden, die Reihenfolge der prints

$$\begin{array}{ll} \text{print}(v) & \Theta(1) \\ \text{ist nicht relevant. } OrderX(l) & \mathcal{O}\left(\frac{k-1}{2}\right) \\ & OrderX(r) \quad \mathcal{O}\left(\frac{k-1}{2}\right) \end{array}$$

Das master-Theorem ist nun anwendbar,

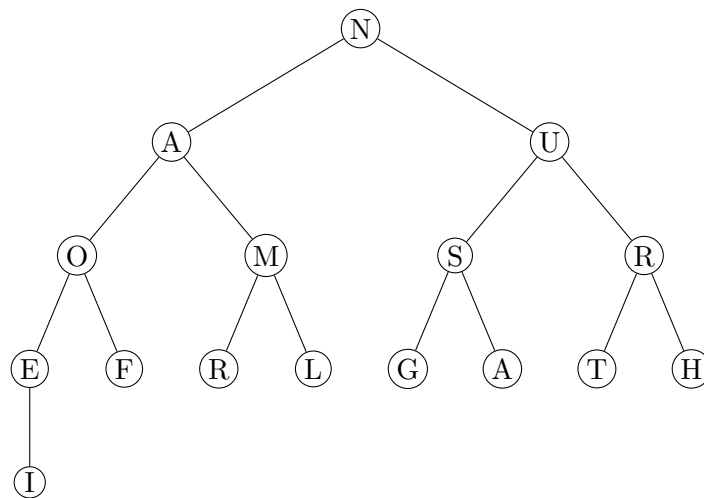
$$T(k) = 2T\left(\left\lceil \frac{k-1}{2} \right\rceil\right) + \mathcal{O}(k^0)$$

Da $\log_2 2 = 1$ gilt, folgt $\mathcal{O}(k^1)$

b)

Die Laufzeiten sind bei gleicher Knotenzahl identische (wie in a) zu sehen, alle Algorithmen haben die gleiche Anzahl an Aufrufen, da nirgends abgebrochen wird, außer wenn keine Kindknoten verfügbar sind).

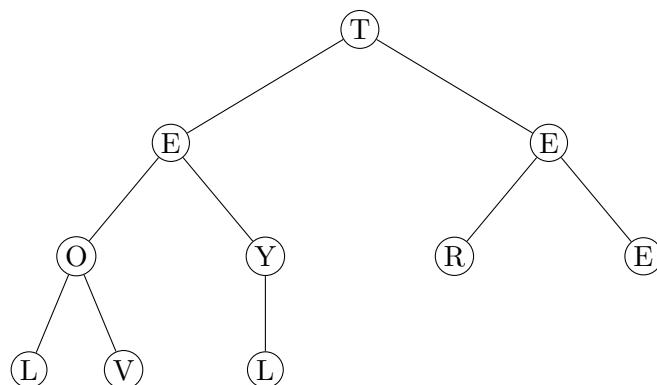
c)



Order1:	N	A	O	E	I	F	M	R	L	U	S	G	A	R	T	H
Order2:	I	E	O	F	A	R	M	L	N	G	S	A	U	T	R	H
Order3:	I	E	F	O	R	L	M	A	G	A	S	T	H	R	U	N

d)

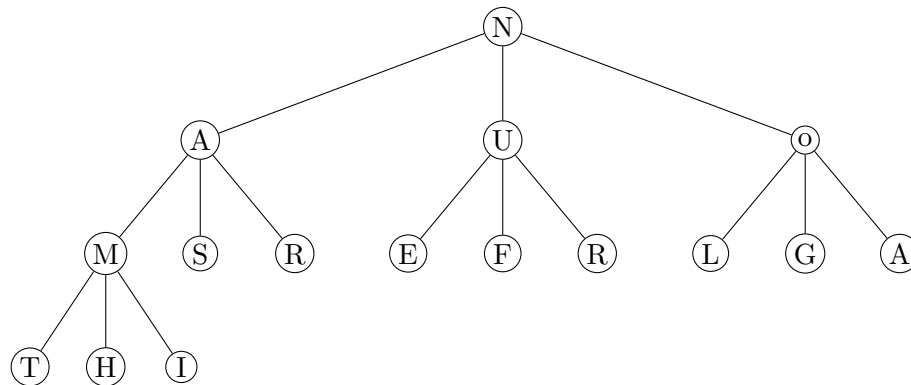
Der LOVELYTREE nach Order 2:

Nach Level-Order: Order3:

T	E	E	O	Y	R	E	L	V	L
---	---	---	---	---	---	---	---	---	---

e)

Ternärer Baum mit vorgegebener Befehlsreihenfolge:

Ausgabe: Order3:

A	L	G	O	R	I	T	H	M	S	A	R	E	F	U	N
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

3. a)

TODO

b)

TODO

c)

TODO

d)

TODO

e)

TODO

f)

TODO

4. a)

TODO

b)

TODO

c)

TODO

5. a)

TODO

b)

TODO