

Übungsblatt 2

Die Abgabe des Blattes erfolgt bis **Montag 12 Uhr** der Folgewoche auf Ilias.

Aufgabe 1: Master Theorem (1+1+1 Punkte)

Die folgenden Terme beschreiben die rekursiven Laufzeiten diverser Divide & Conquer Algorithmen. Wenden Sie das Master Theorem an, um diese Rekurrenzen asymptotisch aufzulösen, wobei e die Eulersche Zahl bezeichnet ($e \approx 2.72$).

$$(a) T(n) = 4 \cdot T(n/4) + n \quad (b) T(n) = T(n/42) + \sqrt[3]{n} \quad (c) T(n) = 10^{11} \cdot T(n/1000) + n^e$$

Aufgabe 2: k -närer Baum (1+1+1+1 Punkte)

In einem k -nären Baum hat jeder Knoten bis zu k Kinder. Der Wurzelknoten liegt in Level $\ell = 0$. Beantworten Sie mit kurzer Begründung:

- Wie viele Kanten hat ein k -närer Baum der Höhe ℓ mit insgesamt n Knoten?
- Wie viele Knoten liegen maximal in Level $\ell \geq 0$?
- Wie viele Knoten hat ein voller Baum ('full Tree') der Höhe ℓ insgesamt?
- Wie viele Knoten hat ein vollständiger Baum ('complete Tree') der Höhe ℓ mindestens?

Aufgabe 3: Tree-Walk (1+3+1+1 Punkte)

Ein vollständiger Binärbaum kann mit der sogenannten "Level-Order" dargestellt werden. Die Knoten eines Baumes werden Level für Level von links nach rechts in ein Array geschrieben. Begonnen wird mit der Wurzel. Das könnte wie folgt aussehen:

6	3	9	2	1	7
---	---	---	---	---	---

. Betrachten Sie nun die folgenden drei Funktionen. Jede erhält als Eingabe die Wurzel eines binären Baumes der Größe n (d.h. mit n Knoten):

```
function PRTORDER1( $v$ )
  if  $v = \text{null}$  then return
  else
    PRINT( $v$ )
    PRTORDER1( $\text{leftChild}$ )
    PRTORDER1( $\text{rightChild}$ )
  end if
end function
```

```
function PRTORDER2( $v$ )
  if  $v = \text{null}$  then return
  else
    PRTORDER2( $\text{leftChild}$ )
    PRINT( $v$ )
    PRTORDER2( $\text{rightChild}$ )
  end if
end function
```

```
function PRTORDER3( $v$ )
  if  $v = \text{null}$  then return
  else
    PRTORDER3( $\text{leftChild}$ )
    PRTORDER3( $\text{rightChild}$ )
    PRINT( $v$ )
  end if
end function
```

- Bestimmen Sie die worst-case Laufzeiten in Abhängigkeit von n . Die PRINT-Operation benötige dabei Zeit $\Theta(1)$. Begründen Sie kurz Ihre Antwort.
- Betrachten Sie den folgenden mit Buchstaben gelabelten Binärbaum, gegeben in "Level-Order":

S	R	M	I	G	O	K	T	H	I	E	L	W	A	L	E
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

. Die PRINT-Operation gebe den Buchstaben des an sie übergebenen Knotens aus. In welcher Reihenfolge werden die Buchstaben jeweils durch den Aufruf von PRTORDER1, PRTORDER2 und PRTORDER3 ausgegeben?
- Bestimmen Sie einen mit Buchstaben gelabelten vollständigen Binärbaum T , für den ein Aufruf von PRTORDER2 die Ausgabe "UNIVERSITY" erzeugt. Geben Sie Ihr Ergebnis in der üblichen Array-Schreibweise (Level-Order von T) an.
- Betrachten Sie nun auf dem Array von Aufgabenteil (c) den zugehörigen ternären Baum, also einen 3-nären Baum, in "Level-Order". Starten Sie darauf eine verzwickte Variante von PRTORDER3 für ternäre Bäume, welche vor dem PRINT-Aufruf die rekursiven PRTORDER3-Aufrufe auf rightChild , leftChild , middleChild in dieser Reihenfolge (!) vornimmt. Was ist nun die Ausgabe?

Bonusaufgabe 1 (3 Punkte) Entwickeln Sie einen Algorithmus, der eine Zahl in einem sortierten (!) Array sucht. Gehen Sie dabei nach dem Divide&Conquer-Prinzip vor. Stellen Sie eine Rekursionsgleichung für die Laufzeit des Algorithmus auf und lösen Sie diese auf. Messen Sie dabei als Laufzeit die Anzahl der benötigten Vergleiche.