

Aufg. 2 Mastertheorem

a)

$$T(n) = 4T\left(\frac{n}{4}\right) + 5$$

$$a=4 \quad b=4 \quad f(n)=5$$

$$n^{\log_b a} = n^{\log_4 4} = n$$

$$f(n)=5 \in O(n^{\epsilon-1}) \text{ mit } \epsilon=1 > 0$$

 \Rightarrow Fall 1

$$T(n) = \Theta(n)$$

b)

$$T(n) = 4T\left(\frac{n}{2}\right) + 2n^2$$

$$a=4 \quad b=2 \quad f(n)=2n^2$$

$$n^{\log_2 4} = n^2$$

$$f(n) = 2n^2 \in \Theta(n^2)$$

 \Rightarrow Fall 2

$$T(n) = \Theta(n^2 \cdot \log n)$$

Bsp:

$$T(n) = 3T\left(\frac{n}{27}\right) + \sqrt[3]{n+1}$$

$$a=3 \quad b=27 \quad f(n) = \sqrt[3]{n+1}$$

$$n^{\log_{27} 3} = n^{\frac{1}{3}} = \sqrt[3]{n}$$

$$f(n) = \sqrt[3]{n+1} \in \Theta(n^{\log_{27} 3}) = \Theta(\sqrt[3]{n})$$

 \Rightarrow Fall 2

$$T(n) = \Theta(\sqrt[3]{n} \log n)$$

Erklärung:

Fall 1

Falls gilt:

$$f(n) \in O(n^{\log_b a - \epsilon}) \text{ mit } \epsilon > 0$$

Dann:

$$T(n) = \Theta(n^{\log_b a})$$

(Nebenkosten vernachlässigbar)

Fall 2

Falls gilt:

$$f(n) \in \Theta(n^{\log_b a})$$

Dann:

$$T(n) = \Theta(n^{\log_b a} \cdot \log(n))$$

(Nebenkosten \approx unterprobleme)Fall 3:

Falls gilt:

$$f(n) = \Omega(n^{\log_b a + \epsilon}) \text{ mit } \epsilon > 0$$

$$a \cdot f\left(\frac{n}{b}\right) \leq c \cdot f(n) \quad \text{für ein } 0 < c < 1 \text{ und hinreichend große } n.$$

Dann:

$$T(n) = \Theta(f(n))$$

Alternative (Annahme $f(n) \in \Theta(n^d)$)

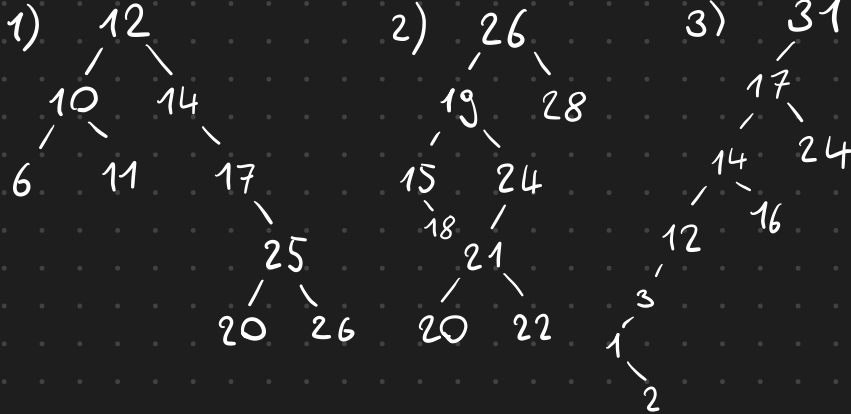
$$T(n) = a \cdot T\left(\frac{n}{b}\right) + \Theta(n^d)$$

mit $a \geq 0$, $b > 1$ und $d \geq 0$

$$T(n) = \begin{cases} \text{Fall 1: } \Theta(n^{\log_b a}) & \text{wenn } d < \log_b a \\ \text{Fall 2: } \Theta(n^d \log n) & \text{wenn } d = \log_b a \\ \text{Fall 3: } \Theta(n^d) & \text{wenn } d > \log_b a \end{cases}$$

Aufg 1. Binäre Suchbäume

a)



b)



a) Zeichnen

Links: kleinere Zahlen

rechts: größere Zahlen

(Tipp: auf alle anwendbar)

b) die Zahl die gelöscht

wird, wird durch die

zahl rechts, ganz links

ersetzt

~~Löschen~~
~~ersatz~~



Aufg 3. Algorithmen

a)

1) false 2) true \Rightarrow Prüft ob ^{min} 2 Ziffern gleich sind

b)

$$n(n-1) \Rightarrow \Theta(n^2)$$

c) Alle Zahlen gleich

d) Alle Zahlen verschieden

e)

i	Ausführungen
1	0
2	1
3	3
4	6
5	10
6	15
n	$\sum_{i=0}^{n-1} i$

$$1 \underline{1} \underline{1} \underline{1}$$

Aufg 4 Greedy Verfahren

b)

Festplatte

1	2	3	4	5
0 1015	1 222 3 360 4 430	2 821 5 407 6 600	7 420	8 800 9 190

(1012) (1020)

a)

c)

Festplatte

1	2	3	4	5
0 1015	2 820 3 190	8 800 1 220	6 600 7 420	4 430 3 360 5 40

d)

Festplatte

1	2	3	4	5
5 40 9 190 1 220 7 360	7 420 4 430	6 600	8 800	2 820

01015 bleibt über

fail

Aufg 5 Wissen

a) Speicherprinzip eines Stacks:

LIFO

b) Absteigend sortiertes Array repräsentiert Max-Heap:

true

c) Quicksort bei bereits sortierten Elementen:

Jedes E 1x

d) Mergesort arbeitet nach Teile und beherrsche?

true

e) Ein rekursiver Algo kann durch welches verfahren optimiert werden?:

Dynamische Programmierung

Aufg 6 Heapsort

Initial 50 | 30 | 5 | 20 | 10 | 40

Build Max Heap 50 | 30 | 40 | 20 | 10 | 5

Max Heapify 40 | 30 | 5 | 20 | 10 | 50

Max Heapify 30 | 20 | 5 | 10 | 40 | 50

Max Heapify 20 | 10 | 5 | 30 | 40 | 50

... 10 | 5 | 20 | 30 | 40 | 50

... 5 | 10 | 20 | 30 | 40 | 50

