

4. Test (Sortieren) - Gruppe A

Name	
Matrikelnummer	
Punkte	

1. Aufgabe (3 Punkte)

Die Laufzeit T eines rekursiven Algorithmus kann für ein Problem der Größe n durch folgende Rekursionsformel bestimmt werden:

$$T(n) = 1 \cdot T\left(\frac{n}{3}\right) + 5$$

Für $n = 1$ beträgt die Laufzeit $\mathcal{O}(1)$.

Bestimmen Sie mit Hilfe des Mastertheorems die Laufzeit. Begründen Sie Ihre Antwort ausführlich.

$$a=1, b=3, f(n) \in \Theta(1)$$

$$\text{berechne } n^{\log_b(a)} \stackrel{\text{einsetzen}}{=} n^{\log_3(1)} = n^0 = 1$$

$$\Rightarrow \text{Fall 2 trifft zu}$$

$$\Rightarrow \text{Laufzeit } T(n) \in \Theta(n^{\log_b(a)} \cdot \log_2(n)) = \Theta(\log_2(n))$$

2. Aufgabe (2 Punkte) Stellen Sie den Verlauf des **Aufteilungsschrittes** von Quicksort (ohne die nachfolgenden Rekursionen) für folgende Zahlenfolge dar: 1, 6, 2, 8, 3, 7, 5. Dabei soll als Pivotelement das Element am rechten Rand gewählt werden. Stellen Sie die Zahlenfolge nach jedem Austauschschritt dar.

1, 6, 2, 8, 3, 7, 5

1, 3, 2, 8, 6, 7, 5

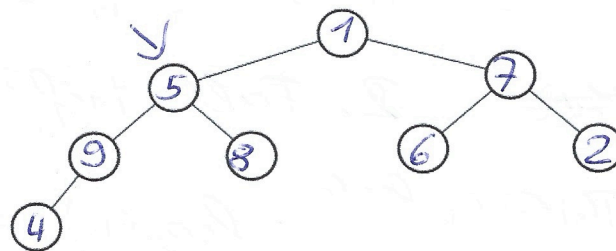
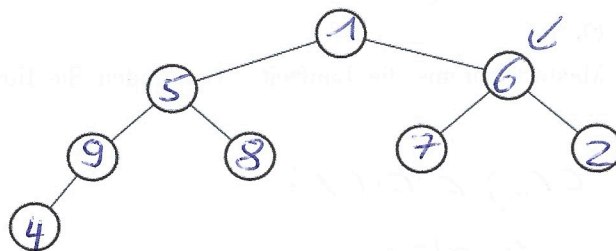
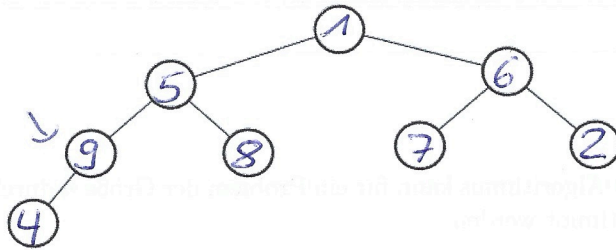
1, 3, 2, 5, 6, 7, 8

} 2

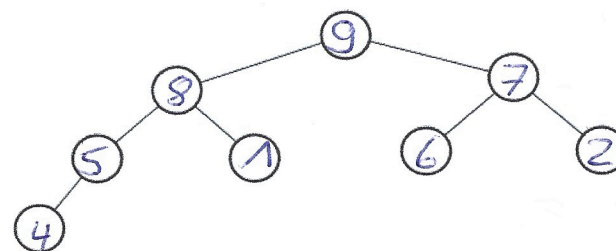
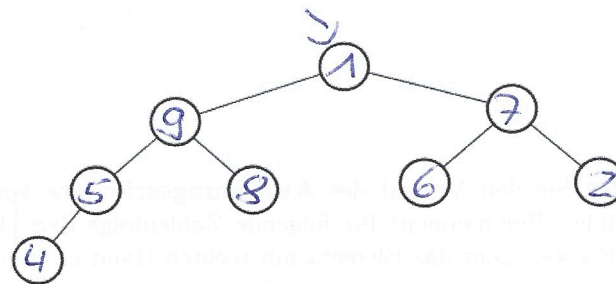
Bitte wenden!

A

3. Aufgabe (5 Punkte) Stellen Sie den Verlauf des Verfahrens BuildHeap zum Aufbau eines Heaps für die Zahlenfolge 1, 5, 6, 9, 8, 7, 2, 4 mit Hilfe der unten aufgeführten Folge von Binärbäumen dar. Geben Sie zur Darstellung des Verfahrens jeweils an, auf welchen Knoten Sie *Heapify* anwenden und anschließend das Ergebnis nach vollständiger Abarbeitung (inklusive aller Rekursionen) von *Heapify*. Das Sortieren mit Hilfe des Heaps ist nicht darzustellen.



(je 11)



4. Test (Sortieren) - Gruppe B

Name	
Matrikelnummer	
Punkte	

1. Aufgabe (3 Punkte)

Die Laufzeit T eines rekursiven Algorithmus kann für ein Problem der Größe n durch folgende Rekursionsformel bestimmt werden:

$$T(n) = 1 \cdot T\left(\frac{n}{4}\right) + 3$$

Für $n = 1$ beträgt die Laufzeit $\mathcal{O}(1)$.

Bestimmen Sie mit Hilfe des Mastertheorems die Laufzeit. Begründen Sie Ihre Antwort ausführlich.

$a=1, b=4, f(n) \in \Theta(1)$
 a, b einsetzen
 berechne $n^{\log_b(a)} \stackrel{!}{=} n^{\log_4(1)} = n^0 = 1$

wg. $f(n) \in \Theta(1) \Rightarrow$ mittlerer Fall trifft zu

$\Rightarrow$ Laufzeit $T(n) = \Theta\left(\underbrace{n^{\log_b(a)}}_{1 \text{ (so)}} \cdot \log_2(n)\right)$
 $= \Theta(\log_2(n))$

2. Aufgabe (2 Punkte) Stellen Sie den Verlauf des **Aufteilungsschrittes** von Quicksort (ohne die nachfolgenden Rekursionen) für folgende Zahlenfolge dar: 2, 6, 1, 8, 4, 7, 5. Dabei soll als Pivotelement das Element am rechten Rand gewählt werden. Stellen Sie die Zahlenfolge nach jedem Austauschschritt dar.

2, 6, 1, 8, 4, 7, 5
 $\uparrow \quad \uparrow$

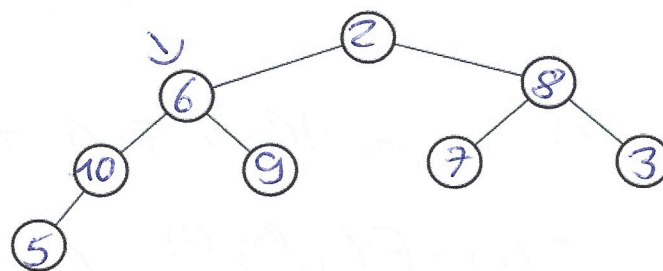
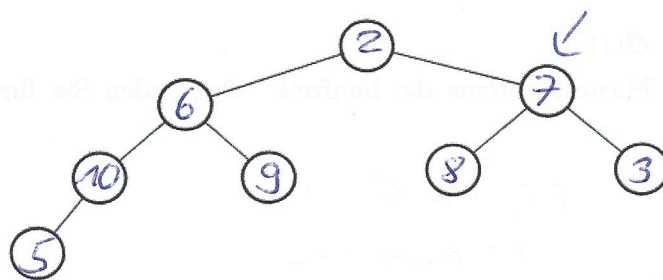
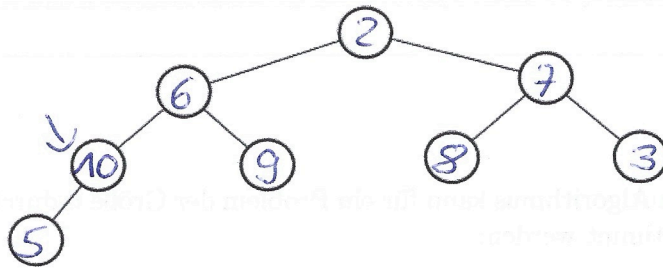
2, 4, 1, 8, 6, 7, 5
 $\times$

2, 4, 1, 5, 6, 7, 8

Bitte wenden!

B

3. Aufgabe (5 Punkte) Stellen Sie den Verlauf des Verfahrens BuildHeap zum Aufbau eines Heaps für die Zahlenfolge 2, 6, 7, 10, 9, 8, 3, 5 mit Hilfe der unten aufgeführten Folge von Binärbäumen dar. Geben Sie zur Darstellung des Verfahrens jeweils an, auf welchen Knoten Sie *Heapify* anwenden und anschließend das Ergebnis nach vollständiger Abarbeitung (inklusive aller Rekursionen) von *Heapify*. Das Sortieren mit Hilfe des Heaps ist nicht darzustellen.



je (1)

