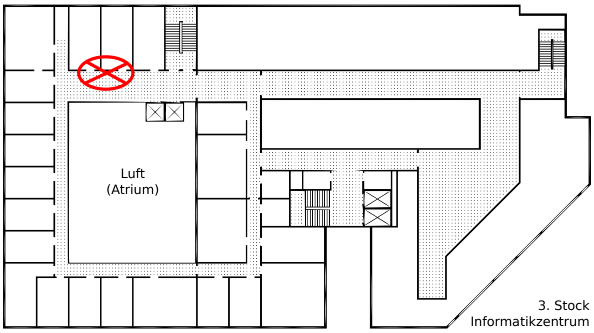
Abteilung Algorithmik **Winter 2017/18**
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig

Prof. Dr. Sándor P. Fekete
Christian Rieck
Arne Schmidt

Algorithmen und Datenstrukturen
Übung 5 vom 15. 01. 2018

Abgabe der Lösungen bis zum Montag, den 29. 01. 2018 um 11:30 im Hausaufgabenschrank bei Raum IZ 337. Es werden nur mit einem dokumentenechten Stift (kein Rot!) geschriebene Lösungen gewertet. **Bitte die Blätter zusammenheften und vorne deutlich mit eigenem Namen, Matrikel- und Gruppennummer, sowie Studiengang versehen!**



Aufgabe 1 (Klausurvorbereitung): **(2 Punkte)**
Gib deinen Namen (Format: Nachname, Vorname), Matrikelnummer, Studiengang und angestrebten Abschluss *leserlich* (in Druckschrift) an. Diese Angaben brauchen wir für die Weiterleitung der Klausurergebnisse, also gebt euch bitte Mühe ;-) .

Aufgabe 2 (Heaps): **(12+6+4 Punkte)**
Algorithmus 1 zeigt einen Sortieralgorithmus. Wir wollen diesen nun untersuchen.

a) Wende Sort(A) auf das Array $A = [3, 1, 5, 2, 6, 4, 7, 8]$ an. Gib das Array nach jedem nicht-rekursiven Aufruf von Max-Heapify an (dazu zählen auch jene in der Funktion Build-Max-Heap). Gib außerdem die Baumstruktur des Heaps nach Build-Max-Heap an. (*Hinweis:* Heaps wurden in der 13. Vorlesung behandelt. Ein Feld $A[i]$ besitzt als Kinder die Felder $A[2i]$ und $A[2i + 1]$.)

b) Zeige, dass Sort(A) ein korrektes Sortierverfahren ist.

c) Zeige, dass Sort(A) eine Laufzeit von $O(n \log n)$ besitzt.

```
function Sort(A)
    Build-Max-Heap(A)
    for i ←length(A) downto 2 do
        vertausche A[1] und A[i]
        heap-größe[A] ← heap-größe[A] − 1
        Max-Heapify(A, 1)
    end for
end function
```

Algorithmus 1: Ein Sortieralgorithmus

Aufgabe 3 (Mastertheorem I): **(3+3+4 Punkte)**
Bestimme die Laufzeiten der folgenden Funktionen mithilfe des in der Vorlesung vorgestellten Mastertheorems. Gib dabei alle im Mastertheorem auftretenden Parameter an.

a)
$$T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{n}{9}\right) + 17\sqrt{n} + T\left(\frac{n}{36}\right) + 57$$

b)
$$U(n) = 7n^2 \log n - 17n^2 + 7 \cdot U\left(\frac{n}{2}\right) + 4n^3 + 7 \cdot U\left(\frac{n}{4}\right)$$

c)
$$V(n) = 2 \cdot V\left(\frac{n}{2}\right) + 1$$

Aufgabe 4 (Mastertheorem II): **(2+2+2 Punkte)**
Auf welche der folgenden Funktionen kann das Mastertheorem aus der Vorlesung angewendet werden? Falls das Mastertheorem angewendet werden kann, gib die Laufzeit der Funktion an. Falls nicht, begründe deine Antwort!

a) $F(n) = F(n-1) + F(n-2)$

b) $W(n) = n + \sum_{i=1}^{3} W(\frac{n}{i})$

c) $X(n) = n^{-1} + 2X(\frac{n}{2})$

Aufgabe 5 (Mergesort): **(15+5 Punkte)**
Gegeben sei folgendes Array:

$$A = [5, 10, 9, 33, 15, 6, 42, 13, 22, 11, 1, 51, 4, 3, 45, 20]$$

a) Gib separat und in chronologischer Reihenfolge die Ergebnisse aller Mergeschritte an, indem Du die Zeilen der Tabelle 1 ausfüllst. Die Mergeschritte auf einem Teilarray der Länge 1 müssen nicht angegeben werden. (Hinweis: Ein Mergeschritt im rechten Teilarray wird erst vorgenommen, wenn das linke Teilarray komplett sortiert worden ist.)

b) In Abbildung 1 ist der Rekursionsbaum der Ausführung von Mergesort aus Aufgabenteil a) abgebildet. Jeder Knoten korrespondiert zu genau einem Mergeschritt. Zeichne in jedem Knoten die Zeilennummer des zugehörigen Mergeschrittes ein. (Hinweis: Die Rekursionstiefe entspricht der Tiefe im Baum.)

$A =$	5	10	9	33	15	6	42	13	22	11	1	51	4	3	45	20
1. $A =$																
2. $A =$																
3. $A =$																
4. $A =$																
5. $A =$																
6. $A =$																
7. $A =$																
8. $A =$																
9. $A =$																
10. $A =$																
11. $A =$																
12. $A =$																
13. $A =$																
14. $A =$																
15. $A =$																

Tabelle 1: Tabelle für Mergesort

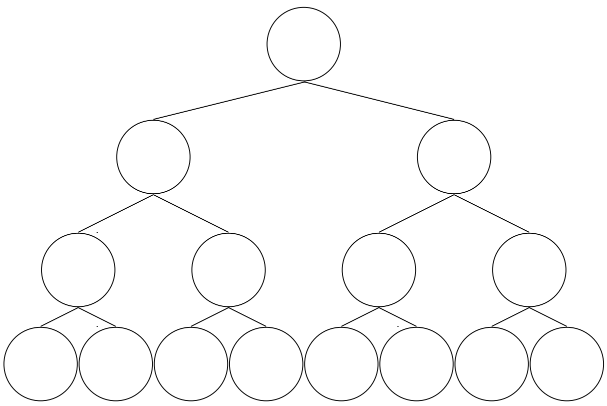


Abbildung 1: Der Rekursionbaum der Ausführung von Mergesort auf A