

# Übungsserie Nr. 12

Ausgabe: 20. Mai 2015

Abgabe: 27. Mai 2015

## Hinweise

Für diese Serie benötigen Sie das Archiv

<http://vs.inf.ethz.ch/edu/FS2015/I2/downloads/u12.zip>.

Aufgrund des Reversi-Turniers am 27. Mai 2015 wird es keine Nachbesprechung zu dieser Serie geben. Sprechen Sie sich stattdessen bei Bedarf individuell mit Ihrem Übungsgruppenleiter ab. Die Themen dieser Serie sind trotzdem prüfungsrelevant.

## 1. Aufgabe: (12 Punkte) Heapsort

(1a) (2 Punkte) Wie viel Elemente sind in einem Heap der Höhe  $h$  mindestens und wie viele sind maximal enthalten?

*Hinweis:* Ein einzelner Knoten ist ein Heap der Höhe 1.

(1b) (2 Punkt) Repräsentiert ein sortiertes Array einen gültigen Heap? Ist die Array-Repräsentation eines Heaps immer sortiert? Begründen Sie jeweils Ihre Antwort!

(1c) (4 Punkte) Sortieren Sie das Array in Abbildung 1 aufsteigend mittels Heapsort. Geben Sie dazu nach jedem Schritt den Inhalt des Arrays an. Wie in der Vorlesung besprochen wird der Heap im Array selbst (“in place”) auf- und abgebaut.

*Hinweis:* Zeichnen Sie zu jedem Schritt den Heap-Baum und lesen Sie daraus die jeweilige Array-Repräsentation ab.

(1d) (4 Punkte) Implementieren sie die Schnittstelle *ISort* mit einem Heapsort-Algorithmus.

Unsortierte Folge:	2	0	3	4	7	5
Schritt 1:						
Schritt 2:						
Schritt 3:						
Schritt 4:						
Schritt 5:						
Heap:						
Schritt 1:						
Schritt 2:						
Schritt 3:						
Schritt 4:						
Schritt 5:						
Sortierte Folge:						

Abbildung 1: Heapsort

## 2. Aufgabe: (10 Punkte) Parallelisiertes Mergesort

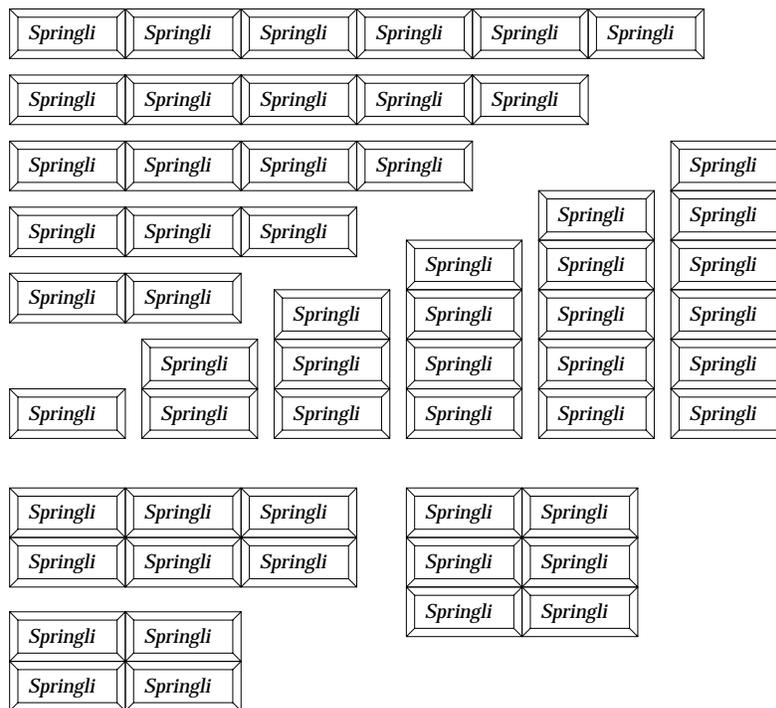
(2a) (8 Punkte) Schreiben Sie eine Implementierung von Mergesort, die eine konfigurierbare Anzahl von Threads verwendet. Dazu wird das zu sortierende Array gleichmässig unter den Threads verteilt, gleichzeitig sortiert und am Ende zu einer Gesamtlösung zusammengeführt.

(2b) (2 Punkte) Schreiben Sie eine Programm, das die verbrauchte Zeit zum Sortieren von 1 Mio. Zufallszahlen misst. Führen Sie die Messungen für eine steigende Anzahl von Threads durch. Wie verändert sich dabei die verbrauchte Zeit? Wie erklären Sie sich das?

## 3. Aufgabe: (6 Punkte) Rekursives Problemlösen

(Nach einer Idee von Prof. Dr. W. Brauer, TU München)

Die Firma *Springli* beabsichtigt, eine neue Schokolade auf den Markt zu bringen. Da über Grösse und Format noch Uneinigkeit herrscht, soll die Marktakzeptanz aller rechteckigen Formate mit maximal  $n$  Stückchen getestet werden. Nachfolgend sind als Beispiel alle 14 möglichen Formate dargestellt, die sich mit maximal 6 Stückchen konstruieren lassen.



Geben Sie die rekursive Funktion  $f(n)$  an, die in Abhängigkeit von  $n$  die Anzahl der Formate, die getestet werden müssen, zurückgibt.

*Hinweis:* Für  $n = 1, 2, 3, 4, 5, 6$  sind 1, 3, 5, 8, 10, 14 Formate zu testen.

#### **4. Aufgabe: (3 Punkte) Lösungsvorschläge zu den Übungen**

Diskutieren Sie, wieso es im Lehrbuch *Marc Allen Weiss, Data Structures and Problem Solving Using Java* Übungsaufgaben gibt, aber keine Lösungen dazu.