

# Übungsserie Nr. 10

Ausgabe: 11. Mai 2011  
Abgabe: 18. Mai 2011

## Hinweise

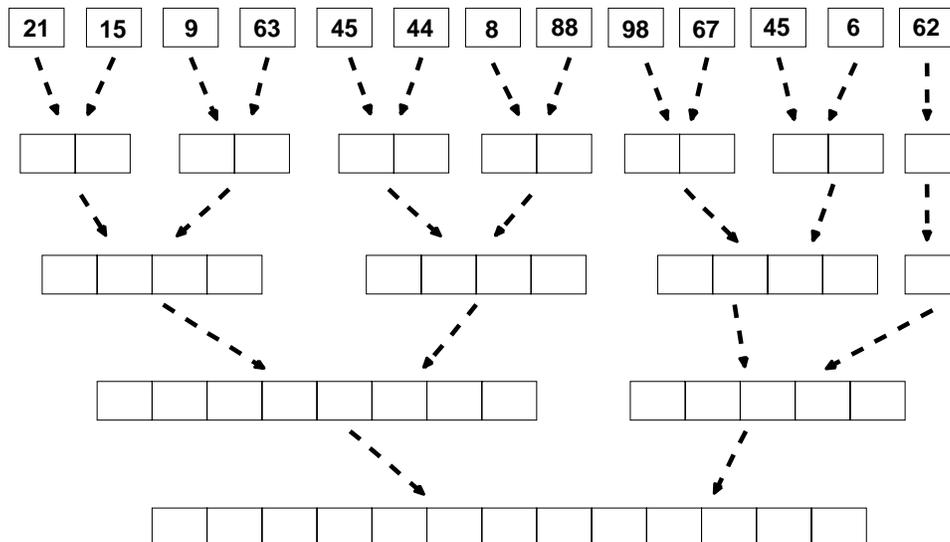
Für diese Serie benötigen Sie die Archive

<http://www.vs.inf.ethz.ch/edu/I2/downloads/u10.zip>

und <http://www.vs.inf.ethz.ch/edu/I2/downloads/reversi.jar>.

## 1. Aufgabe: (10 Punkte) Mergesort

(1a) (2 Punkte) Führen Sie einen Lauf von Mergesort manuell aus. Füllen Sie dazu das folgende Schema. Die erste Zeile enthält eine Sequenz ungeordneter Zahlen, die absteigend sortiert werden soll.



(1b) (4 Punkte) Implementieren Sie die Schnittstelle *ISort* mit einem Mergesort-Algorithmus. Die Fabrikmethode heisst *SortFactory.create*.

(1c) (2 Punkte) Schreiben Sie eine *main*-Klasse, die Ihren Sortieralgorithmus für verschieden grosse, mit zufälligen Zahlen gefüllte Eingaben ausführt und die dafür benötigte Zeit misst. Messen Sie damit die benötigte Zeit zum Sortieren von 100, 200, 400, 800, 1600, 3200, 6400, 12800, 25600 und 51200 Zahlen.

Hinweis: um statistisch stabile Messergebnisse zu erhalten, sollte man jede Messung mehrfach ausführen, den grössten und den kleinsten Wert ignorieren und den Rest mitteln.

(1d) (2 Punkte) Erstellen Sie ein Diagramm mit zwei logarithmischen Achsen für die obigen Messwerte. Überlagern Sie die Messpunkte mit der theoretischen Kurve für die Zeitkomplexität von Mergesort.

Stimmen Ihrer Messergebnisse mit der theoretischen Zeitkomplexität des Mergesort-Algorithmus überein? Begründen Sie Ihre Antwort.

## 2. Aufgabe: (5 Punkte) Türme von Hanoi

In der Vorlesung wurde das Problem der “Türme von Hanoi” vorgestellt und eine rekursive Lösung angegeben.

(2a) (1 Punkt) Für jeden Schritt bei der Ausführung des rekursiven Algorithmus aus der Vorlesung (Folie 10.11) wird genau ein Turm nicht benötigt. Geben Sie für die 15 Schritte bei der Umschichtung eines Turms der Höhe 4 die Folge der Nummern derjenigen Türme an, die *nicht* angefasst werden.

(2b) (2 Punkte) Was fällt Ihnen auf? Können Sie aus dem Ergebnis aus Teilaufgabe a) eine Idee für einen kinderleichten nicht-rekursiven Algorithmus für das Problem entwickeln? Beschreiben Sie diesen Algorithmus in Pseudocode.

(2c) (2 Punkte) Kann man Ihren Algorithmus auch verwenden, falls der Anfangsturm die Höhe 5 besitzt? Begründen Sie Ihre Antwort.

## 3. Aufgabe: (8 Punkte) Reversi [Teil 4]

(3a) (8 Punkte) Implementieren Sie einen Reversi-Spieler, der eine  $\alpha$ - $\beta$ -Analyse des Spielbaumes durchführt. Erhöhen Sie dabei wieder die maximale Rekursionstiefe so lange noch Zeit ist.