

Übungsserie Nr. 10

Ausgabe: 10. Mai 2017
Abgabe: 17. Mai 2017

Hinweise

Für diese Serie benötigen Sie die folgenden Archive:

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

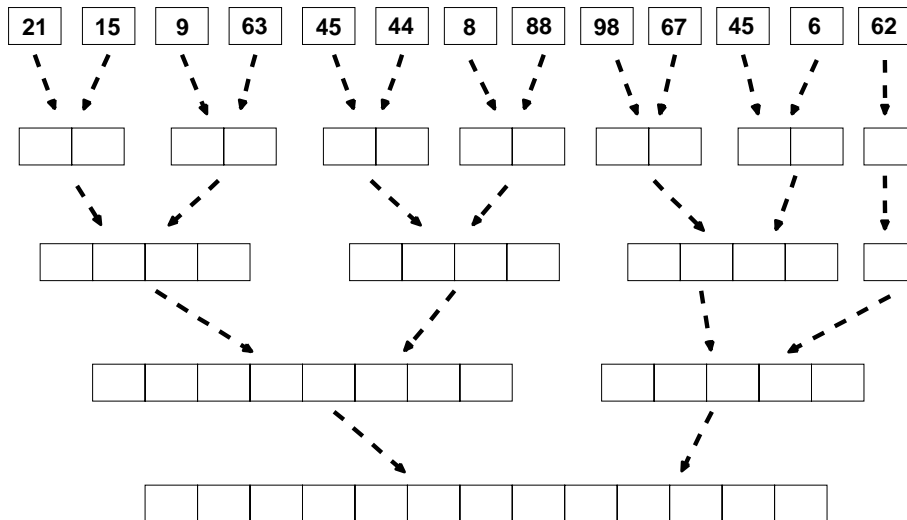
<https://www.vs.inf.ethz.ch/edu/I2/downloads/reversi.jar>.

1. Aufgabe: (10 Punkte) Mergesort

ETH Codeboard für Textaufgaben: <https://codeboard.ethz.ch/ifee2u10a1ad>

ETH Codeboard Programmieren: <https://codeboard.ethz.ch/ifee2u10a1bc>

(1a) (2 Punkte) Führen Sie einen Lauf von Mergesort manuell aus. Nutzen Sie hierfür das folgende Schema. Die erste Zeile enthält eine Sequenz ungeordneter Zahlen, die aufsteigend 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 Ganzzahlen gefüllte Eingaben ausführt. Messen Sie die Zeit, die Ihr Algorithmus zur Sortierung für 100, 200, 400, 800, 1600, 3200, 6400, 12800, 25600 und 51200 Zahlen benötigt.

Hinweis: Um statistisch stabile Messergebnisse zu erhalten, sollte man jede Messung mehrfach ausführen, den grössten und den kleinsten Wert ignorieren und den Durchschnitt der anderen Werte berechnen.

(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 Ihre Messergebnisse mit der theoretischen Zeitkomplexität von Mergesort überein? Begründen Sie Ihre Antwort.

2. Aufgabe: (5 Punkte) Türme von Hanoi

ETH Codeboard: <https://codeboard.ethz.ch/ifee2u10a2>

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 dem Skript (Folie 952 ff.) 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 sehr einfachen 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]

ETH Codeboard: <https://codeboard.ethz.ch/ifee2u10a3>

(3a) (8 Punkte) Implementieren Sie einen Reversi-Spieler, der eine α - β -Analyse des Spielbaumes durchführt. Erhöhen Sie dabei wie in der letzten Reversi-Aufgabe schrittweise die maximale Rekursionstiefe und greifen Sie kurz vor Ende der zur Verfügung stehenden Zeit auf das Ergebnis des letzten vollständigen Durchlaufs zurück.