

Aufgabe 12.1

Um die Parallelverarbeitungsmöglichkeiten moderner Mehrkernprozessoren für eigene Programme zu nutzen, gibt es die Möglichkeit, unabhängige Berechnungen in sogenannten Threads zu organisieren. Diese Threads werden dann, soweit möglich, parallel ausgeführt. Im Material zur Vorlesung sind die Klassen

- `MySort.java` und
- `SortingThreadLauncher.java`

enthalten. Als Beispiel soll hier die parallele Sortierung mehrerer Arrays von ganzen Zahlen mit InsertionSort betrachtet werden.

Die Klasse `SortingThreadLauncher.java` enthält bereits eine `main`-Methode, die eine gewünschte Anzahl von Threads initialisiert und startet. Das Programm wird dann durch einen Aufruf der Form

```
java SortingThreadLauncher.java 4 100000
```

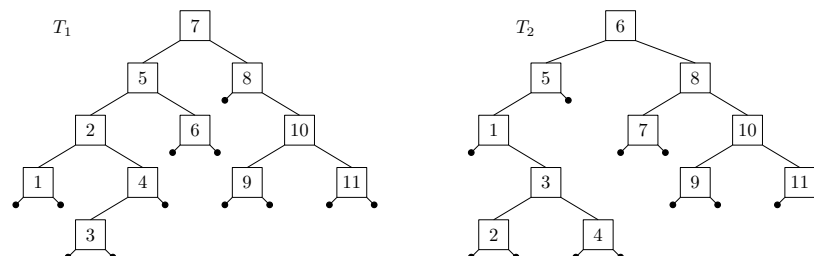
gestartet. Bei diesem konkreten Aufruf würden 4 Arrays mit jeweils 100000 Elementen sortiert. Dabei erfolgt die Sortierung jedes Arrays in einem eigenen Thread.

Der noch fehlende Baustein ist der Code, der innerhalb eines solchen Threads ausgeführt werden soll, also das Sortieren mit InsertionSort. Dieser Code ist in der Klasse `MySort.java` zusammengefasst, ist aber noch unvollständig. Insbesondere erwartet der Konstruktor der Klasse `Thread` einen Parameter vom Typ `Runnable`. Dabei handelt es sich um ein Interface und dieses soll von der Klasse `MySort.java` geeignet implementiert werden. Vervollständigen Sie die Klasse `MySort.java` entsprechend.

(4 Punkte)

Aufgabe 12.2

Geben Sie eine Folge von Rotationen an, mit denen Sie den unten abgebildeten binären Suchbaum T_1 in den ebenfalls unten abgebildeten binären Suchbaum T_2 überführen können.



(2 Punkte)

Aufgabe 12.3

Versuchen Sie zu argumentieren, dass man allgemein für jedes Paar (T_1, T_2) von binären Suchbäumen, in denen die gleiche Schlüsselmenge X verwaltet wird, eine Folge von Rotationen angeben kann, die T_1 in T_2 überführt.

(4 Punkte)