

Präsenzaufgaben 5

26./27.04.2021

Die Lösung der Aufgaben wird am Ende der Übung von Ihnen vorgestellt.

Aufgabe 1

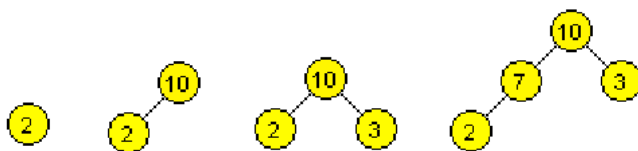
Fügen Sie in der Klasse `BinarySearchTree` (s. Veranstaltungsseite) folgende rekursive Methoden ein:

- **public** `String toString()`: Gibt die Werte des Suchbaums in In-Order-Reihenfolge (bzw. aufsteigender Reihenfolge) zurück. Siehe Vorlesungsvideo. Verbessern Sie im Vergleich zum Vorlesungsvideo die Behandlung der Kommas.
- **public int** `getElementCount()`: Siehe Vorlesungsvideo
- **public int** `getSum()`: Siehe Vorlesungsvideo
- **public int** `getHeight()`: Gibt die Höhe des Baums zurück.
- **public int** `getLeafCount()`: Gibt die Anzahl der Blätter des Baums zurück.
- **public boolean** `hasNodesWithOneChild()`: Gibt zurück, ob der Baum Knoten mit einem einzigen Kind besitzt.

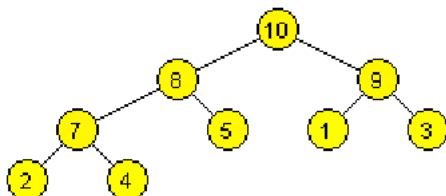
Aufgabe 2

- a) Fügen Sie die folgenden Werte nacheinander in einen Heap ein. Zeichnen Sie nach jedem Einfügen den Heap neu.
2,10,3,7,5,1,9,4,8

Hinweis: Die ersten Schritte sind:



- b) Entfernen Sie vom folgenden Heap nacheinander alle Werte, bis der Heap aufgelöst ist. Zeichnen Sie nach jedem Entfernen den Heap neu.



Aufgabe 3

Die Funktion `getLargestKElements(int[] list, int k)` aus der Klasse `HeapUtils` (s. Veranstaltungsseite) gibt ein Feld mit den `k` größten Elementen aus dem Feld `list` zurück. Der Code ist zwar kurz, aber trotzdem anspruchsvoll und natürlich vollkommen unkommentiert.

- In main steht ein kurzes Testprogramm. Finden Sie anhand der Kontrollausgaben und der „Erklärung“ des Codes auf der Seite <https://www.geeksforgeeks.org/k-largestor-smallest-elements-in-an-array/> heraus, wie der Algorithmus funktioniert. Es handelt sich auf der Seite um Methode 6.
- Kommentieren Sie alle schwierigen Codestellen (also den gesamten Code) der Funktion `getLargestKElements`. Ihre Kommentare sollten unter anderem folgende Punkte enthalten:
 - Was bedeutet der Konstruktoraufruf von `q`? Wie müsste man ihn ändern, um die `k` kleinsten Elemente zu finden?
 - Was beinhaltet der Heap `q` nach der ersten Schleife?
 - Was beinhaltet der Heap `q` nach jedem Durchlauf der 2. Schleife?
 - Was passiert genau in der `return`-Zeile?
 - Welche O-Klasse hat die Funktion? Begründen Sie Ihre Antwort. Achten Sie auf die O-Klassen der Methoden `add` und `poll`.