
Grundlagen: Algorithmen und Datenstrukturen

Abgabetermin: In der jeweiligen Tutorübung

Hausaufgabe 1

Implementieren Sie in der Klasse `UIaHeap` einen adressierbaren binären Heap, der neben den Basisoperationen auch die Operationen `insert`, `remove` und `decreaseKey` implementiert.

Verwenden Sie für Ihre Implementierung die auf der Übungswebseite bereitgestellten Klassen und verändern Sie für Ihre Implementierung *ausschließlich* die Klasse `UIaHeap`.

Achten Sie bei der Abgabe Ihrer Aufgabe darauf, dass Ihre Klasse `UIaHeap` heißt und auf den Rechnern der Rayhalle (`rayhalle.informatik.tu-muenchen.de`) mit der bereitgestellten Datei `main_ah` kompiliert werden kann. Anderenfalls kann eine Korrektur nicht garantiert werden. Achten Sie darauf, dass Ihr Quelltext ausreichend kommentiert ist.

Schicken Sie die Lösung per Email mit dem Betreff `[GAD] Gruppe <Gruppennummer>` an Ihren Tutor.

Hinweis: Hier macht die Benutzung eines Dynamischen Arrays Sinn. Falls Sie die früher benutzte Klasse `UIsArray` benutzen wollen, so fügen Sie den Quellcode Ihrer Abgabe hinzu. Sie können aber auch das in Java implementierte Dynamische Array benutzen, welches deutlich flexibler ist.

Aufgabe 1

Führen Sie auf einem anfangs leeren Binären Heap folgende Operationen aus und stellen Sie die Zwischenergebnisse graphisch dar:

- `build({15, 20, 9, 1, 11, 8, 4, 13, 17})`,
- `insert(7)`,
- `delMin()`,
- `decreaseKey(20, 3)`,
- `delete(8)`.

Aufgabe 2

Analysieren Sie die Anzahl der Element-Vergleiche, die der in der Vorlesung angegebene `siftDown`-Algorithmus im Worst-Case benötigt.

Geben Sie ausgehend von dem Algorithmus in der Vorlesung einen Algorithmus an, der mit $\log n + \mathcal{O}(\log \log n)$ Vergleichen auskommt.