

Aufgabenblatt 3 Stand: 30.11.2023

Bearbeiten Sie alle Aufgaben aller Übungsblätter – soweit nicht anders angegeben – in festen **Zweiergruppen**. Beide Gruppenmitglieder müssen die Aufgabe gemeinsam bearbeiten und die Lösung alleine erläutern können.

Aufgabe 3.1 – Binary Tree

Im MS Teams Raum wurde Ihnen ein Interface für einen Binary Tree bereitgestellt. Implementieren Sie dieses in einer eigenen Klasse und sichern Sie Ihre Implementierung mit geeigneten Tests ab. Achten Sie auch auf die Absicherung der Traversierungen. **Sie werden diese Klasse in Aufgabe 4 wiederverwenden, daher macht es noch mehr Sinn die Implementierung mit guten Tests abzusichern!**

Aufgabe 3.2 – Binary Search Tree (BST)

Auch für einen BST wurde Ihnen ein Interface bereitgestellt, welches Sie implementieren sollen. Hierfür gibt es allerdings auch schon eine vorbereitete Klasse BSTree.java mit einigen wenigen Tests, die Sie erweitern sollen. **Sie werden auch diese Klasse in Aufgabe 4 wiederverwenden, daher macht es noch mehr Sinn die Implementierung mit guten Tests abzusichern!**

Aufgabe 3.3 – Empirische Analyse Binary Search Tree

Analysieren Sie Ihren BST, indem Sie diesen wie folgt mit zufälligen Schlüsselwerten füllen. Die Anzahl der Schlüsselwerte soll von $n = 100$ bis $n = 10.000$ reichen (in 100er Schritten, also 100 Messwerte). Für jede Schlüsselanzahl wiederholen Sie die nachfolgende Messung (Zufall!) 1.000 Mal. Messen Sie jeweils die durchschnittliche Pfadlänge eines zufälligen Knoten und stellen Sie das Arithmetische Mittel Ihrer Messungen dar. Plotten Sie in Ihrem Diagramm auch die Funktion $1,39 \cdot \log_2 n - 1,85$. Wie verhält sich Ihr BST im Verhältnis zu der angegebenen Funktion?

Hinweis: Die durchschnittliche Pfadlänge eines beliebigen Knoten ermitteln Sie, indem Sie die interne Pfadlänge durch die Größe des Baumes teilen und um eins erhöhen. Die interne Pfadlänge ergibt sich aus der Summe aller Pfadlängen der einzelnen Knoten.

Aufgabe 3.4 – Graph

Implementieren Sie eine generische Datenstruktur für einen Graphen in einer der beiden in der Vorlesung vorgestellten Varianten.

Für „Ihren“ Graphen implementieren Sie die Algorithmen:

1. Tiefensuche
2. Breitensuche
3. Kürzeste Pfade nach Dijkstra

Nehmen Sie die Beispiele zu den Algorithmen aus der Vorlesung als **minimale(!)** Testbasis. Passt die erwartete Laufzeit des Dijkstra-Algorithmus zu ihrer Implementierung? Wie haben Sie dies überprüft?

Abgabe

Ihre Ergebnisse fassen Sie in einer Dokumentation zusammen und geben diese per Email ab.

Die Dokumentationen enthalten:

- Titel, die Namen der Autoren, die Praktikumsgruppe, Teamnummer und ein Datum.
- Ein Abstract (kurze Zusammenfassung der Arbeit)

Die Dokumentationen sind üblicherweise gegliedert in:

- Einführung. Hier wird in den Themenbereich eingeführt.

- Beschreibung von Verfahren, Implementationen, etc. Dies umfasst insbesondere auch UML-Diagramme und Auszüge (! keine Kopien !) aus dem Quelltext.
- Validierung und Tests. Hier wird dargestellt wie die Funktionsweise der Lösung geprüft wurde.
- Ggf. Referenzen. Die benutzten Quellen (das betrifft auch Quellcode), die im Text referenziert werden.

Weitere Hinweise:

- Schreiben Sie in wissenschaftlich neutralem Stil
- Grafiken sind zu nummerieren. Achsen in quantitativen Darstellungen sind zu beschriften. Angabe der Einheit (ms, ns, ...) beachten.
- Werden Grafiken genutzt, sind diese auch im Text zu erläutern und zu referenzieren.

Die Dokumentation gehen wir im Praktikum durch. Im Anschluss an das Praktikum senden Sie mir Ihre Dokumentation als **PDF-Datei** per Email mit folgenden Betreff zu (der Betreff ist gleichzeitig auch der **Dateiname**):

[BITS3-AD] Abgabe Aufgabe 2 Praktikumsgruppe X – Team Y

Die Werte für X und Y ersetzen Sie entsprechend.

Die Abgabe **muss bis zum Ende des Tages an dem Ihr Praktikum stattfindet** erfolgen.

Viel Spaß bei der Umsetzung!

Lars Hamann