

Einführung in die Programmierung

Prof. Dr. Franziska Matthäus

Prof. Dr. Matthias Kaschube

Dr. Karsten Tolle



Übungsblatt ÜE-06

Ausgabe: 29.11.2023

Abgabe: 09.12.2023

13:00 Uhr

Datenstrukturen

Hinweise:

- Es dürfen keine Lösungen aus dem Skript, dem Internet oder anderen Quellen abgeschrieben werden. Diese Quellen dürfen nur mit Quellenangaben verwendet werden und es muss ein hinreichend großer Eigenanteil in den Lösungen deutlich zu erkennen sein.
- Digitale Abgaben, die nicht im Format **.pdf** oder **.txt** für Texte oder **.py** für Code erfolgen, werden nicht bewertet. Bei Abgaben mehrerer Dateien müssen diese als **.zip** zusammengefasst werden.
- Achten Sie darauf die Variable **__author__** in allen Quellcode Dateien (.py) korrekt zu setzen (am Anfang des Quellcodes): **__author__ = "<Matr-Nr>, <Nachname>"**
Beispiel: **__author__ = "1234567, Tolle"**
... Leerstellen vor und nach dem „=" und Leerstelle nach dem Komma beachten, sowie keine spitzen Klammern verwenden.
- Außerdem muss Ihr Name in jeder abgegebenen **.pdf** und **.txt** Datei zu finden sein. Abgaben der Dokumentation, die per Hand geschrieben und eingescannt werden, sind nur in zuvor abgesprochenen Ausnahmefällen erlaubt.
- Datei- und Ordnernamen sollen keine Umlaute, diakritische oder Sonderzeichen, mit Ausnahme des Unterstrichs in der Namensmitte, enthalten!

Aktualisierung der Hinweise:

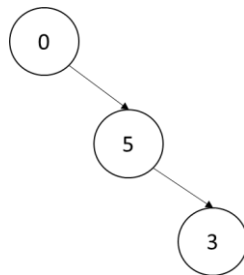
- Beim Programmieren und Kommentieren halten Sie sich an die Regeln im Programmierhandbuch, siehe Moodle-Kurs ([Programmierhandbuch WiSe 23/24 \(Style Guide\)](#)). Im Zweifelsfall gilt PEP 8.

Σ 10 Punkte

Aufgabe 1 (ehemalige Klausuraufgabe!)

5 Punkte

1.1 Geben Sie für die beiden dargestellten gewurzelten Binärbaume die Traversierung in Prä- und Inorder-Reihenfolge an. Geben Sie weiterhin an, ob der Baum eine HEAP-Eigenschaft erfüllt (bitte ankreuzen) und begründen Sie ihre Antwort (ohne richtige Begründung gibt es keine Punkte).



Präorder:

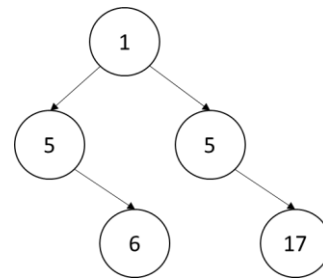
0, 5, 3

Inorder:

0, 5, 3

HEAP-Eigenschaft: Ja ☐ Nein ☒

Begründung:



Präorder:

1, 5, 6, 6, 17

Inorder:

5, 6, 1, 5, 17

HEAP-Eigenschaft: Ja ☒ Nein ☐

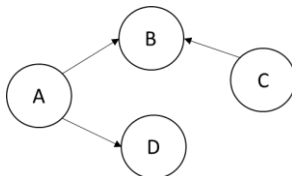
Begründung:

Erster Graph: Es liegt hier weder die MAX-Heap noch die Min-Heap Eigenschaft vor, da der Knoten mit dem maximalsten Wert in der mittlersten Ebene liegt.

Zweiter Graph: Hier liegt die Min-Heap Eigenschaft vor, da in der obersten Ebene der minimalste Wert und in der untersten Ebene die höchsten Werte liegen.

1.2 Geben Sie zu den folgenden Graphen an, ob es sich um Bäume handelt oder nicht. Listen Sie weiterhin auf, welche der Knoten Blätter und welche Wurzeln. Sollte es keine Blätter- oder Wurzel-Knoten geben, so geben Sie die leere Menge {} an. Sollte etwas nicht eindeutig sein, so begründen Sie Ihre Entscheidung:

a)

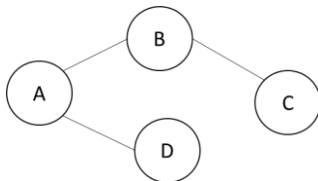


Baum: Ja ☒ Nein ☐

Blätter: D, B

Wurzel: {}

b)



Baum: Ja ☒ Nein ☐

Blätter: C, D

Wurzel: {}

c)



Baum: Ja ☒ Nein ☐

Blätter: A

Wurzel: {}

Aufgabe 2

5 Punkt

Schreiben Sie ein Python 3.X Programm, mit welchem man Knoten und Kanten zwischen diesen eingeben kann. Hierbei gehen wir davon aus, dass es sich um einen gerichteten Graphen handelt. Das Programm soll ausgeben:

- a) Ob es sich bei dem eingegebenen Graphen um einen Baum handelt oder nicht.
- b) Wenn es ein Baum ist, sollen die Blätter ausgegeben werden.
- c) Weiterhin soll geprüft werden, ob der Baum eine Wurzel hat und wenn ja, welcher Knoten dies ist.

Trennen Sie das Hauptprogramm von eventuell verwendeten Funktionen, so dass diese in einem eigenen Modul ausgelagert und dort getestet werden können. Erstellen Sie für das Hauptprogramm eine entsprechende Dokumentation.