

Universität Potsdam
Institut für Informatik
Algorithmen und Datenstrukturen

08. Aufgabenblatt

Lernziele (zum Abhaken): Die Student:innen können...

für eine vorgegebene sowie zu erfüllende Eigenschaft eines Baumes (Anzahl der Knoten, Blätter sowie inneren Knoten, Tiefe) einen zugehörigen Baum angeben.

einen vollständigen ungerichteten Graphen angeben.

für einen gegebenen Graphen zugehörige aufspannende Bäume angeben, die nicht durch Drehung oder Spiegelung auseinander hervorgehen.

elementare Eigenschaften von Bäumen selbstständig beweisen.

die verschiedenen Methoden für die Traversierung von Bäumen (Inorder, Preorder, Postorder und Levelorder) innerhalb von Python unter der Benutzung der Schnittstelle ADT Baum implementieren.

13 Bäume

1. Zeichnen Sie je einen binären Baum der Tiefe 3
 - (a) mit minimaler Anzahl von Knoten
 - (b) mit maximaler Anzahl von Knoten
 - (c) mit minimaler Anzahl von Blättern
 - (d) mit maximaler Anzahl von inneren Knoten.
2. Ein ungerichteter Graph heist *vollstndig*, falls seine Adjazenzmatrix, abgesehen von der Hauptdiagonalen, nur Einsen enthlt, d.h. falls paarweise verschiedene Knoten immer durch eine Kante miteinander verbunden sind.
 - (a) Zeichnen Sie einen vollstndigen ungerichteten Graphen G mit 4 Knoten.
 - (b) Zeichnen Sie vier verschiedene aufspannende Bume von G , die nicht durch Drehen oder Spiegeln auseinander hervorgehen.
3. Beweisen Sie folgende Aussagen:
 - (a) Ein ungerichteter Graph besitzt einen aufspannenden Baum gdw. er zusammenhngend ist. Zeigen Sie auch, dass zu jedem ungerichteten Graphen ein aufspannender Baum konstruiert werden kann.
 - (b) (*durch vollstndige Induktion:*) Ein binrer Baum der Tiefe d hat hchstens $2^{d+1} - 1$ viele Knoten.
4. Implementieren Sie die Methoden Preorder, Postorder und Levelorder als Python-Funktionen. Arbeiten Sie mit der Schnittstelle des ADT Baum aus der Vorlesung, d.h. Sie drfen die Funktionen dieser Schnittstelle aufrufen.