Philipps-Universität Marburg
Fachbereich Mathematik und Informatik
Arbeitsgruppe Datenbanksysteme

Sommersemester 2023
19. 06. 2023

Prof. Dr. Bernhard Seeger
Jana Holznigenkemper M.Sc.
Marius Kuhrt M.Sc.

Übungen zur Vorlesung
Algorithmen und Datenstrukturen

Abgabe: 25. 06. 2023,
bis **spätestens** 23:59 Uhr
über die ILIAS Plattform

Übung 10

Aufgabe 10.1: Eigenschaften von Rot-Schwarz-Bäumen (1+1+3) (5 Punkte)

Betrachten Sie im Folgenden Rot-Schwarz-Bäume.

a) Zeigen Sie: Alle Pfade, die von der Wurzel zu Blättern führen, haben dieselbe Anzahl an schwarzen Knoten.

b) Wenn der kürzeste Pfad von der Wurzel zu einem Blatt die Länge l_{min} hat, wie groß kann dann die Länge l_{max} des längsten Pfades von der Wurzel zu einem Blatt maximal sein?

c) Wie viele Knoten hat ein Rot-Schwarz-Baum der Höhe h mindestens? Begründen Sie Ihre Antwort.

Aufgabe 10.2: Implementierung von Graphen (4+2+3+6*) (15 Punkte)

Im Ilias finden Sie das Interface `Graph<T>`, welches einen Graphen repräsentiert, und eine Implementierung dieses Interfaces `AdjacencyListGraph<T>`, welche einen Graphen durch Adjazenzlisten realisiert.

a) Implementieren Sie das Interface `Graph<T>` durch eine Klasse `AdjacencyMatrixGraph<T>` (siehe Ilias), welche einen Graphen mit Hilfe einer Adjazenzmatrix verwaltet. Die Matrix soll mit einer Größe von 10×10 Feldern initialisiert werden und wachsen, sobald diese Kapazität überschritten wird.

b) Erweitern Sie Ihre Klasse `AdjacencyMatrixGraph<T>` um eine Methode `convert()`, welche den Graphen in Adjazenzlistenrepräsentation überführt, d.h. ein Objekt der Klasse `AdjacencyListGraph<T>` zurückliefert. Implementiseren sie die entsprechende Methode `AdjacencyListGraph<T>`. Geben Sie die Laufzeiten Ihrer Implementierungen an.

c) Implementieren Sie in der Klasse `AdjacencyListGraph<T>` die Methode `undirected()`, die eine neue Repräsentation des Graphen erstellt, dessen Kanten ungerichtet sind. Achten Sie darauf, dass dieser neue Graph auch bei weiteren Änderungen ungerichtet bleibt.

d) Ein Graph heißt *bipartit*, wenn die Knoten des Graphen mit genau 2 Farben so gefärbt werden können, dass Knoten, welche über eine Kante verbunden sind, unterschiedlich Farben besitzen. Entwickeln Sie einen Algorithmus, welcher auf einem ungerichteten Graphen überprüft, ob dieser bipartit ist. Beschreiben Sie Ihren Algorithmus zunächst textuell. Implementieren Sie Ihren Algorithmus anschließend in der Klasse `AdjacencyListGraph<T>` durch eine Methode `undirectedGraphIsBipartit()`. Hierbei soll der Graph zuerst in eine ungerichtete Repräsentation überführt werden, auf welcher danach die Überprüfung stattfindet.

*** Aufgabe 10.2 d) ist für Lehramtsstudierende optional.**