

Algorithmen und Datenstrukturen Klausur SS 2006

Software-Engineering und Technische Informatik Bachelor

Die Klausur besteht aus 7 Aufgaben und umfasst 60 Punkte. Als Hilfsmittel sind alle schriftlichen Unterlagen gestattet. Viel Erfolg!

Aufgabe 1 AVL-Baum

(10 Punkte)

Fügen Sie in einem leeren AVL-Baum die Zahlen

7, 6, 5, 1, 3, 9, 8

ein und löschen Sie dann die Zahlen

1, 5, 3.

Stellen Sie die Binärbäume dar, die sich als Zwischenschritte ergeben. Geben Sie die durchgeführten Rotationsoperationen und die zuvor vorhandenen Höhenunterschiede für die relevanten Knoten an.

Aufgabe 2 AVL-Baum

(6 Punkte)

Student Peter behauptet, dass in einem AVL-Baum der Höhe h sich jedes Blatt in der Ebene h oder $h-1$ befindet. Widerlegen Sie Peter durch ein Gegenbeispiel.

Aufgabe 3 Hashverfahren

(8 Punkte)

Geben Sie die Belegung einer Hash-Tabelle der Größe $n = 7$ an, wenn folgende Operationen durchgeführt werden:

- Einfügen: 4, 5, 32, 1, 8
- Löschen: 5
- Einfügen: 22

Verwenden Sie offenes Hashing mit quadratischem Sondieren und der Hash-Funktion

$$h(k) = k \bmod n.$$

Geben Sie für jeden Einfügevorgang die Sondierungsfolge an.

Aufgabe 4 Hashverfahren

(10 Punkte)

Die etwa 8000 Mitarbeiter eines Unternehmens mit mehreren Standorten in Deutschland sollen mit einer Hashtabelle verwaltet werden. Als Schlüssel wird die Personalnummer gewählt, die folgenden Aufbau besitzt:

ttmmjjjjaaappppp

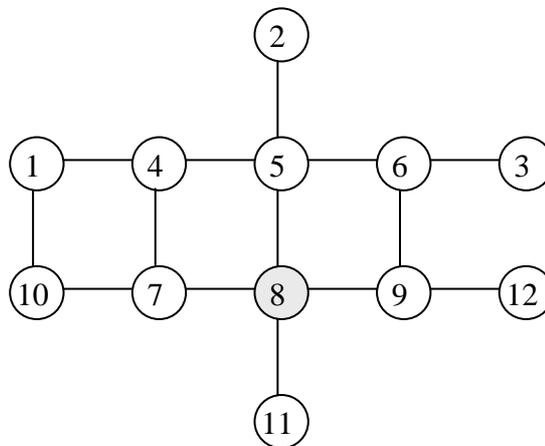
„ttmmjjjj“ stellt das Geburtsdatum dar, bestehend aus Tag, Monat und Jahr. „aaa“ gibt die Abteilung an, der der Mitarbeiter zugeordnet ist, und „ppppp“ die Postleitzahl des Standorts. Es soll ein offenes Hashverfahren mit quadratischem Sondieren gewählt werden.

- Welches Problem entsteht, wenn als Tabellengröße $n = 10000$ und die übliche Hashfunktion $h(k) = k \bmod n$ gewählt wird.
- Geben Sie eine Hashfunktion an (C++-Code; Schlüssel k als String), die das Problem aus a) vermeidet. Die Tabellengröße $n = 10000$ soll dabei unverändert bleiben.
- Wie lässt sich das Problem aus a) vermeiden, falls die Tabellengröße leicht verändert werden darf?
- Treffen Sie eine Aussage über den Aufwand, einen Mitarbeiter in der Tabelle zu suchen. Unterscheiden Sie zwischen erfolgreicher und nicht erfolgreicher Suche. Gehen Sie dabei von einer Hashfunktion wie in b) oder c) beschrieben aus.

Aufgabe 5 Tiefen- und Breitensuche in Graphen

(8 Punkte)

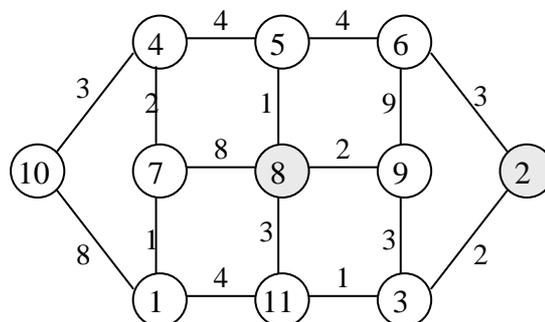
In welcher Reihenfolge werden die Knoten besucht, falls bei Knoten 8 beginnend der Graph mit Tiefen- bzw. Breitensuche durchlaufen wird. Gehen Sie davon aus, dass für jeden Knoten die Nachbarknoten nach ihrer Nummer in aufsteigender Reihenfolge betrachtet werden.



Aufgabe 6 Kürzeste Wege in Graphen

(12 Punkte)

Bestimmen Sie mit dem Algorithmus von Dijkstra den kürzesten Wege von Knoten 8 nach 2. Geben Sie nach jedem Besuchsschritt die Liste der Kandidaten, die als nächstes besucht werden können, mit ihren aktuellen Distanzen zum Startknoten an. Wie sieht die vollständige Datenstruktur zur Speicherung der kürzesten Wege aus, nachdem der kürzeste Weg gefunden wurde? Geben Sie den kürzesten Weg an.



Aufgabe 7 Minimal aufspannender Baum

(6 Punkte)

Bestimmen Sie für den Graphen aus Aufgabe 6 mit dem Algorithmus von Kruskal den minimal aufspannenden Baum. Es genügt, wenn Sie den minimal aufspannenden Baum angeben. Welche Kantengewichtssumme hat er?