Prof. Dr. Ir. Joost-Pieter Katoen

Christian Dehnert, Jonathan Heinen, Thomas Ströder, Sabrina von Styp

Hinweise:

- Die Übungsblätter sind in Gruppen von je 3 Studierenden aus der gleichen Kleingruppenübung zu bearbeiten.
- Die Lösungen müssen bis Montag, den 11. Juni um 11:00 Uhr in den entsprechenden Übungskasten eingeworfen werden. Sie finden die Kästen am Eingang Halifaxstr. des Informatikzentrums (Ahornstr. 55).
- Namen und Matrikelnummern der Studenten sowie die Nummer der Übungsgruppe sind auf jedes Blatt der Abgabe zu schreiben. Heften bzw. tackern Sie die Blätter!

Aufgabe 1 (Countingsort): (4 + 2 Punkte)

a) Das folgendes Array ist mit Countingsort zu sortieren. Geben Sie das Histogramm- und das Positionsarray vor dem ersten Einfügen ins Ausgabearray an, sowie das Positions- und Ausgabearray nach jedem Einfügeschritt.

4	3	0	1	4	2	3	7	3

b) Der in der Vorlesung vorgestellte Algorithmus Countingsort fügt die Elemente des Eingabearrays von hinten nach vorne in das Ausgabearray ein. Welche Nachteile ergäben sich, wenn man das Eingabearray stattdessen von vorne nach hinten durchlaufen würde?

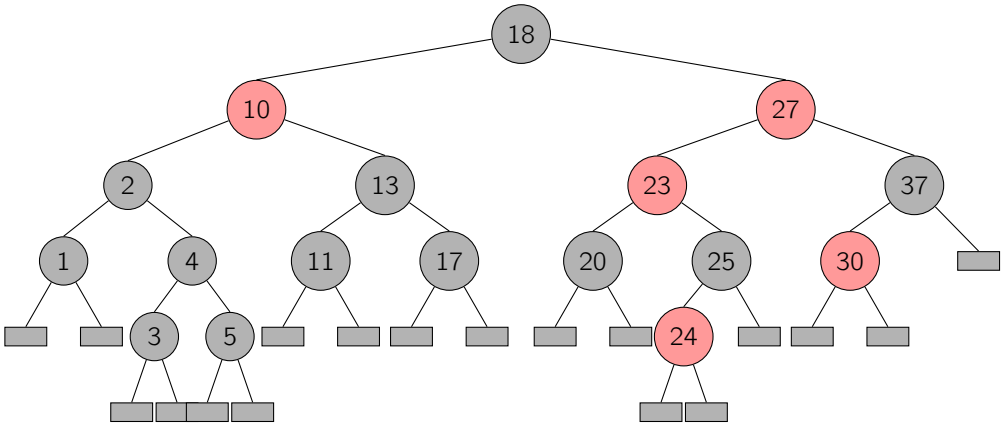
Aufgabe 2 (Rot-Schwarz Bäume): (6 + 4 Punkte)

a) Fügen Sie die folgenden Werte in der gegebenen Reihenfolge in einen leeren Rot-Schwarz-Baum ein:

$$12, 7, 19, 5, 6, 23, 25, 21, 15, 20$$

Zeichnen Sie den Baum jeweils nach dem Einfügen und nach jeder erfolgten Rotation und Umfärbung. Machen Sie außerdem kenntlich, welche Transformation in welchem Schritt passiert.

b) Gegeben sei der folgende Baum:



Handelt es sich um einen Rot-Schwarz-Baum? Ist dies nicht der Fall, stellen Sie die Rot-Schwarz-Eigenschaft durch **Umfärben** von **maximal zwei** Knoten her.

Löschen Sie nun die folgenden Werte in der gegebenen Reihenfolge aus ihrem (ggf. "reparierten") Rot-Schwarz-Baum: 2, 18. Zeichnen Sie den Baum jeweils nach dem Löschen und nach jeder erfolgten Rotation und Umfärbung. Machen Sie außerdem kenntlich, welche Transformation in welchem Schritt passiert.

Aufgabe 3 (Hashing): (3 + 3 + 2 Punkte)

a) Betrachten Sie die folgenden Funktionen, die ein Wort $s = a_1 \dots a_n$ auf einen Hash-Wert zwischen 0 und m abbilden. Dabei sind die a_i als ASCII-Werte gegeben, also $0 \leq a_i \leq 127$ für alle i.

- $h_1(s) = n \mod m$
- $h_2(s) = (\sum_{k=1}^{n} a_k) \mod m$
- $h_3(s) = (\sum_{k=1}^{n} k \cdot a_k) \mod m$
- $h_4(s) = ((h_{good}(s)^{p-1} \mod p) \mod m)$

wobei $h_{good}(s)$ eine Hash-Funktion ist, die alle wichtigen an eine Hash-Funktion gestellten Eigenschaften erfüllt und p eine Primzahl ist. Diskutieren Sie, inwiefern die Funktionen h_i $(1 \leq i \leq 4)$ als Hash-Funktionen geeignet sind.

b) Gegeben sei eine Hash-Table der Größe m und eine beliebige Hash-Funktion $h\colon U \to \{0, \dots, m-1\}$. Die Menge U habe nun mindestens $n \cdot m$ Elemente, also $|U| \geq n \cdot m$. Zeigen Sie, dass U eine Teilmenge U_0 der Größe n besitzt ($|U_0| = n$), so dass

$$h(x_1) = h(x_2) \quad \text{für alle} \quad x_1, x_2 \in U_0$$

Was haben Sie damit für die Worst-Case-Laufzeit der Suche mittels Hashing mit Verkettung bewiesen?

c) Gegeben sei nun eine Hash-Table mit einer initialen Größe von 1000 und eine Hash-Funktion, die ein gleichverteiltes Hashing gewährleistet. Nach wie vielen Einfügungen müssen sie a-priori mit einer Kollisionswahrscheinlichkeit von mehr als 80% rechnen?

Um die Anzahl von Kollisionen beim Hashing gering zu halten, kann man die Größe der Hash-Table nach einer gewissen Anzahl von Einfügungen erhöhen. Nach welcher Anzahl k von Einfügeoperationen muss die Tabelle das erste Mal vergrößert werden, wenn bei den vorhergehenden $k-1$ Einfügeoperationen keine Kollision auftrat und die Wahrscheinlichkeit für eine Kollision bei der k-ten Einfügung weniger als 20% betragen soll? Begründen Sie ihre Antwort.

Aufgabe 4 (Hashing): (3 + 2 + 1 Punkte)

Gegeben sei eine Hash-Table der Größe 23 und die folgenden beiden Hash-Funktionen über der Universalmenge $U = \{0, \dots, 499\}$:

- $h_1(x) =$ Quersumme von x
- $h_2(x) = x \mod 23$

a) Fügen Sie die Werte 12, 99, 111, 76, 23, 30 sowohl mit h_1 als auch h_2 mittels

(i) Hashing mit Verkettung

(ii) Hashing mit linearem Sondieren

in jeweils eine Tabelle ein (es sind also 4 Tabellen zu erstellen). Geben Sie die nicht-leeren Teile der Tabellen nach jedem Einfügeschritt an.

b) Löschen Sie nacheinander die Werte 111, 12 und 76 aus allen Tabellen aus Aufgabe **a)**. Geben Sie die nicht-leeren Teile der Tabellen nach jedem Löschschritt an. Erläutern Sie, welches Vorgehen dafür jeweils nötig ist.

c) Suchen Sie in den Tabellen, die aus Teilaufgabe **b)** resultierten, nach dem Wert 30. Erläutern Sie, welches Vorgehen dafür jeweils nötig ist.