

# Algorithmen und Datenstrukturen I

WS 2009/10, 4. Aufgabenblatt, Abgabe 09.12.2009

## Aufgabe 13

8 Punkte

In dem uns nicht so fernen Land “Formularien” wird jedes Jahr eine Volkszählung durchgeführt. Dazu ist jeder Bürger in jeder Gemeinde verpflichtet! Die ausgefüllten Formulare sollen nach bestehender Amtsanweisung innerhalb jeder Gemeinde nach dem Geburtsdatum sortiert werden. Jedem Gemeindevorsteher steht hierbei offen, ob er sich für das Sortieren nach dem *Bucketsort*-Verfahren oder für ein anderes Sortierverfahren mit  $\Theta(n \log_2 n)$  an Aufwand entscheidet. Es ist von ihm allerdings in jedem Fall sicherzustellen, dass es immer das günstigste Verfahren in Abhängigkeit zur Anzahl der Gemeindebewohner darstellt, da die Mitarbeiter der örtlichen Behörde tunlichst mehr Falschparker notieren sollen, anstatt stupides Sortierwerk zu bewerkstelligen.

Für das Sortieren nach *Bucketsort* stehen in jeder Gemeinde drei Klassen von Waschkörben bereit, die mit  $Y$ ,  $M$  und  $D$  bezeichnet sind und in Quantitäten von  $Y = 100$  Stück,  $M = 12$  Stück und  $D = 31$  Stück geliefert wurden.

- (a) Geben Sie in Abhängigkeit der Gemeindegröße  $n$  an, für welches Sortierverfahren der Gemeindevorsteher sich entscheiden soll. Für *Bucketsort* wird angenommen, dass Duplikate immer erst hinzugefügt werden, wenn jeder einzelne Korb einer Klasse mit mindestens einem Formular belegt wurde. Für die konstanten Kosten für das Streuen und Sammeln wird entsprechend der zusätzliche Aufwand von  $C_1 = 1$  und  $C_2 = 1$  amtlich festgelegt.
- (b) Nennen Sie alternative Lösungsstrategien für das Verwenden von *Bucketsort*, mit den genannten Klassen und Anzahlen an Waschkörben.

## Aufgabe 14

5 Punkte

Sortieren Sie das Array  $A = (6, 38, 14, 25, 4, 14)$  mittels des Ihnen in der Vorlesung vorgestellten Heap-Sort Algorithmus unter Verwendung eines Max-Heaps. Geben Sie alle Zwischenschritte, z.B. das Wiederherstellen der Heap-Eigenschaft, an und markieren Sie die bereits sortierte Teilliste.

## Aufgabe 15

12 Punkte

Gegeben sei die folgende Namensliste:

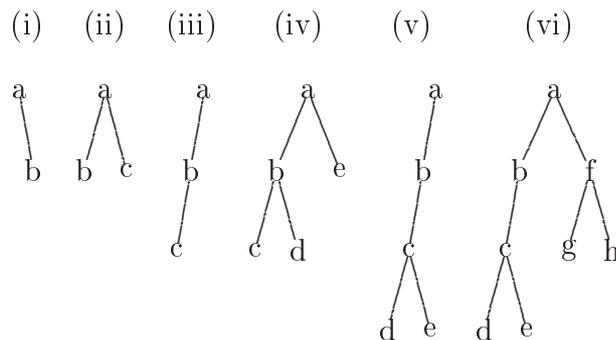
Simpson, Lisa  
Flanders, Ned  
Flanders, Maude  
Simpson, Bart  
Wiggum, Chief  
Simpson, Marge  
Wiggum, Ralph  
Springfield, Jebediah  
Simpson, Homer

- (a) Geben Sie die Liste nach einer Sortierung mit **Selection-Sort**, **Merge-Sort** und **Quick-Sort** an. Bei Quick-Sort werde das jeweils **letzte** Element einer Liste als Pivot-Element benutzt. Schlüssel seien nur die Nachnamen, Vornamen werden bei Schlüsselvergleich ignoriert. Bei welchen der Sortierverfahren können Sie bereits anhand dieses Beispiels erkennen, dass sie nicht stabil sortieren? *Hinweis: Wir empfehlen den in der Vorlesung angegebenen Pseudo-Code tatsächlich zu implementieren und die Listen von Ihren Programmen sortieren zu lassen. Das reduziert die Gefahr den einen oder anderen Flüchtigkeitsfehler zu begehen.*
- (b) Geben Sie einen möglichst einfachen Trick an, durch den jedes auf paarweisen Schlüsselvergleich beruhende Sortierverfahren stabil gemacht werden kann.

## Aufgabe 16

10 Punkte

Gegeben seien die folgenden Binärbäume.



- (a) Geben Sie für jeden der Bäume (i)-(vi) dessen Höhe an, und ob er die folgenden Eigenschaften hat (ja/nein): strikt, ausgeglichen, fast vollständig, vollständig. Geben Sie Ihre Ergebnisse in Form einer Tabelle, in der zu jedem der Bäume jeweils eine Spalte steht. (6 Punkte)
- (b) Geben Sie zu jedem der Bäume (i)-(iv) an, durch welchen Ausdruck er gemäß ADT-Spezifikation BINTREE erzeugt wird. (4 Punkte)