

Aufgabe 2.1 (4 Punkte)

In einer Liste sind die Namen verschiedener Vorlesungen zusammen mit einer natürlichen Zahl von 0 bis 4, die für die veranstaltende Fakultät steht, gespeichert:

(0, Algebra), (3, Computernetze), (1, VWL), (2, Strafrecht),
(3, Betriebssysteme), (1, BWL), (0, Analysis), (4, Anatomie), (0, Topologie)

Sortieren Sie die Datensätze mit Hilfe von Bucket-Sort aufsteigend nach der ersten Komponente. Geben Sie dazu die resultierenden fünf Buckets $Q[0]$ bis $Q[4]$ unmittelbar vor ihrer Entleerung und die resultierende Sortierung der Datensätze an.

Aufgabe 2.2 (4 Punkte)

Sortieren Sie folgende Schlüssel mit Hilfe von Lexicographic Sort:

SINNE, ESSEN, SEIEN, EISEN, SENSE, SEHEN, MEINE, EINEN, SEINE, ESSER.

Geben Sie den Inhalt von `QUEUE` nach jedem Durchlauf der äußeren Schleife an.

Aufgabe 2.3 (4 Punkte)

Sei v ein Knoten des Baumes T , dann gilt:

- $depth(v) :=$ Länge des Pfades von der Wurzel zu v ,
- $height(v) :=$ Länge eines längsten Pfades von v zu einem Blatt,
- $height(T) := height(root(T))$,
- $level(v) := height(T) - depth(v)$.

Zeigen Sie, dass das Level eines jeden Knotens in einem binären Baum in Linearzeit bestimmt werden kann. Geben Sie dazu einen geeigneten Algorithmus an. Anschließend ist

- o die Korrektheit,
- o die Laufzeit

Ihres Algorithmus zu zeigen.

Aufgabe 2.4 (4 Punkte)

In der Vorlesung wurde die Datenstruktur *einfach-verkettete Liste* definiert. Auf Grundlage dieser Datenstruktur definieren wir eine neue Datenstruktur *zirkuläre Liste*, für die der letzte „echte“ Knoten des Box-Pointer-Diagramms auf einen vorhergehenden Knoten zeigt. Hier ein Beispiel:

	A	NEXT
0	–	1
1	4	2
2	2	3
3	0	4
4	9	5
5	7	6
6	6	7
7	8	8
8	1	5

- a) Geben Sie einen Algorithmus an, der prüft, ob die übergebene verkettete Liste zirkulär ist. Dieser darf die Arrays A und NEXT nicht verändern und darüber hinaus nur einen konstanten Platzbedarf haben. Der Zeitbedarf muss linear in der Anzahl der Knoten sein. Sie dürfen voraussetzen, dass alle Werte im Array A unterschiedlich sind.
Vorsicht: Der Algorithmus kennt *nicht* die Anzahl n der Knoten, so dass es nicht ausreicht zu testen, ob man nach n Schritten das Ende der Liste erreicht.
- b) Skizzieren Sie einen Korrektheitsbeweis Ihres Algorithmus.
- c) Zeigen Sie, dass Ihr Algorithmus den vorgegebenen Platz- und Zeitbedarf hat.

Abgabe: Lösungen können jeweils bis zum folgenden Dienstag um 12:00 Uhr in die Kästen vor NA 02/257 (Nähe Rechenzentrum Servicecenter) *nach Aufgaben getrennt* eingeworfen werden. Geben Sie ihren Namen, ihre Matrikelnummer und ihre Gruppe an. Auf jedem abgegebenen Aufgabenzettel dürfen bis zu drei Namen stehen.