

Übung zur Vorlesung Datenstrukturen und Algorithmen

Eigenständige Präsenzübung (Gruppe A)

Name: _____

Matrikelnummer: _____

Alle Antworten sind zu beweisen!

Aufgabe 1 (10 Punkte)

```
class Listnode<K, D> {  
    K key;  
    D data;  
    Listnode<K, D> pred, succ;  
    Listnode(K k, D d) {  
        key = k; data = d; pred = null;  
        succ = null;  
    }  
    void delete() {  
        pred.succ = succ; succ.pred = pred;  
    }  
    void append(Listnode<K, D> newnode) {  
        newnode.succ = succ;  
        newnode.pred = this;  
        succ = newnode;  
    }  
}
```

In der Vorlesung wurde eine Beispielimplementierung für eine doppelt verkettete Liste vorgestellt. Betrachten Sie den Quellcode links für die Klasse eines zugehörigen Listenknotens, welcher leider einen Fehler enthält: Wenn n ein bereits in der verketteten Liste enthaltener Listenknoten ist, dann sollte ein Aufruf $n.append(\text{new Listnode}\langle K, D\rangle(\text{key}, \text{value}))$ eigentlich einen neuen Listenknoten mit den gegebenen Werten key und $value$ hinter den aktuellen Knoten n in die Liste einfügen, wie es das Bild unten veranschaulicht. Leider ist die Implementation fehlerhaft, so daß der Aufruf von $append$ nicht das gewünschte tut. Erklären Sie, wo der Fehler liegt, und wie man ihn beheben kann.

Liste vorher:

$$\dots \Leftrightarrow (k_1, v_1) \Leftrightarrow (k_2, v_2) \Leftrightarrow (k_3, v_3) \Leftrightarrow \overbrace{(k_4, v_4)}^n \Leftrightarrow (k_5, v_5) \Leftrightarrow \dots$$

Gewünschtes Ergebnis:

$$\dots \Leftrightarrow (k_1, v_1) \Leftrightarrow (k_2, v_2) \Leftrightarrow (k_3, v_3) \Leftrightarrow \overbrace{(k_4, v_4)}^n \Leftrightarrow \overbrace{(key, value)}^{\text{neues Element}} \Leftrightarrow (k_5, v_5) \Leftrightarrow \dots$$

Übung zur Vorlesung Datenstrukturen und Algorithmen

Eigenständige Präsenzübung (Gruppe B)

Name: _____

Matrikelnummer: _____

Alle Antworten sind zu beweisen!

Aufgabe 1 (10 Punkte)

Gegeben sei ein Array a mit n Elementen, welches folgende Eigenschaft hat: Es ist derzeit nicht sortiert, läßt sich jedoch durch den *einmaligen* Tausch zweier Elemente i, j in eine sortierte Form überführen. Man muß also nur einmal $swap(i, j)$ auf a aufrufen, um ein sortiertes Array zu erhalten.

Leider wissen Sie nicht, um welche Elemente i, j es sich handelt. Entwerfen Sie einen Algorithmus, der für ein gegebenes Array mit n Elementen, welches obige Eigenschaft hat, in $O(n)$ Schritten feststellt, welche zwei Elemente i, j vertauscht werden müssen, so daß a danach sortiert ist. Sie dürfen Pseudocode oder richtigen Code angeben oder den Algorithmus umgangssprachlich beschreiben. Begründen Sie jedoch ausführlich, warum Ihr Algorithmus die richtigen Elemente in $O(n)$ Schritten findet.

Übung zur Vorlesung Datenstrukturen und Algorithmen
Eigenständige Präsenzübung (Gruppe C)

Name: _____

Matrikelnummer: _____

Alle Antworten sind zu beweisen!

Aufgabe 1 (10 Punkte)

Beweisen oder widerlegen Sie:

$$n \log n = O((\sqrt{n})^{\log n})$$