

Algorithmen und Datenstrukturen Übung 3

## 1 Aufgabe 3.1

```
public class Stack {
    Element top;

    boolean isEmpty(){
        if (top == null)
            return true;
        else
            return false;
    }

    int top(){
        return top.value;
    }

    int pop(){
        int temp = top();
        top = top.next;
        return temp;
    }

    void push(int v) {
        // Fuege Element hinzu
        if (isEmpty()) {
            Element add = new Element(v);
            top = add;
        } else {
            Element tmp = new Element(v, top);
            top = tmp;
        }
    }

    static class Element {
        int value;
        Element next;

        Element(int v) {
            value = v;
        }

        Element(int v, Element n) {
            this(v);
            next = n;
        }
    }
}
```

```

        }
    }
}

```

## 2 Aufgabe 3.2

**a)** Der Algorithmus schiebt immer die größte Zahl so weit nach oben bis entweder die oberste Zelle erreicht wird oder er eine noch größere Zahl findet und diese dann weiter verschiebt. Nach jeder Durchführung dieses Prozesses wird die oberste Zelle um eine verringert da die 'alte' oberste Zelle jetzt mit der größten möglichen Zahl gefüllt ist. Dies wird dann so lange wiederholt bis das Feld aufsteigend sortiert ist.

**b)** Da das Algorithmus verschachtelte For-Schleifen verwendet ist die Laufzeit im schlimmsten Fall  $O(n^2)$