Aufgabe 4

Gegeben seien die Standardstrukturen Stapel (Stack) und Schlange (Queue) mit folgenden Standardoperationen:

Stapel	Schlange
`boolean isEmpty()`	`boolean isEmpty()`
`void push(int e)`	`enqueue(int e)`
`int pop()`	`int dequeue()`
`int top()`	`int head()`

Beim Stapel gibt die Operation `top()` das gleiche Element wie `pop()` zurück, bei der Schlange gibt `head()` das gleiche Element wie `dequeue()` zurück. Im Unterschied zu `pop()`, beziehungsweise `dequeue()`, wird das Element bei `top()` und `head()` nicht aus der Datenstruktur entfernt.

(a) Geben Sie in Pseudocode einen Algorithmus `sort(Stapel s)` an, der als Eingabe einen Stapel `s` mit `n` Zahlen erhält und die Zahlen in `s` sortiert. (Sie dürfen die Zahlen wahlweise entweder aufsteigend oder absteigend sortieren.) Verwenden Sie als Hilfsdatenstruktur ausschließlich eine Schlange `q`. Sie erhalten volle Punktzahl, wenn Sie außer `s` und `q` keine weiteren Variablen benutzen. Sie dürfen annehmen, dass alle Zahlen in `s` verschieden sind.

```
q := neue Schlange
while s not empty:
    q.enqueue(S.pop())
while q not empty:
    while s not empty and s.top() < q.head():
        q.enqueue(s.pop())
    s.push(q.dequeue)
```

Als Java-Code:

```java
/**
 * So ähnlich wie der <a href=
 * "https://www.geeksforgeeks.org/sort-stack-using-temporary-
↪  stack/">Stapel-Sortiert-Algorithmus
 * der nur Stapel verwendet</a>, nur mit einer Warteschlange.
 *
 * @param s Der Stapel, der sortiert wird.
 */
public static void sort(Stapel s) {
  Schlange q = new Schlange();
  while (!s.isEmpty()) {
    q.enqueue(s.pop());
  }
  while (!q.isEmpty()) {
    // Sortiert aufsteigend. Für absteigend einfach das „kleiner"
    // Zeichen umdrehen.
    while (!s.isEmpty() && s.top() < q.head()) {
      q.enqueue(s.pop());
    }
    s.push(q.dequeue());
  }
}
```

(b) Analysieren Sie die Laufzeit Ihrer Methode in Abhängigkeit von n.

> Zeitkomplexität: $\mathcal{O}(n^2)$, da es zwei ineinander verschachtelte `while`-Schleifen gibt, die von der Anzahl der Elemente im Stapel abhängen.