

Einzelprüfung „Theoretische Informatik / Algorithmen (vertieft)“

Einzelprüfungsnummer 66115 / 2017 / Frühjahr

## Thema 1 / Aufgabe 2

(Top-Level-Domains (TLD))

**Stichwörter:** Sortieralgorithmen, Bucketsort, Radixsort, Mergesort, Quicksort

In dieser Aufgabe sei vereinfachend angenommen, dass sich Top-Level-Domains (TLD) ausschließlich aus zwei oder drei der 26 Kleinbuchstaben des deutschen Alphabets ohne Umlaute zusammensetzen. Im Folgenden sollen TLDs lexikographisch aufsteigend sortiert werden, d. h. eine TLD  $(s_1, s_2)$  mit zwei Buchstaben (z. B. „co“ für Kolumbien) wird also vor einer TLD  $(t_1, t_2, t_3)$  der Länge drei (z. B. „com“) einsortiert, wenn  $s_1 < t_1 \vee (s_1 = t_1 \wedge s_2 \leq t_2)$  gilt.

- (a) Sortieren Sie zunächst die Reihung [„de“, „com“, „uk“, „org“, „co“, „net“, „fr“, „ee“] schrittweise unter Verwendung des Radix-Sortiervorgangs (Bucketsort). Erstellen Sie dazu eine Tabelle wie das folgende Muster und tragen Sie dabei in das Feld „Stelle“ die Position des Buchstabens ein, nach dem im jeweiligen Durchgang sortiert wird (das Zeichen am TLD-Anfang habe dabei die „Stelle“ 1).

### Exkurs: Alphabet

abcdefghijklmnopqrstuvwxyz

Lösungsvorschlag

Stelle	Reihung							
	de_	com	uk_	org	co_	net	fr_	ee_
3	de_	uk_	co_	fr_	ee_	org	com	net
2	de_	ee_	net	uk_	co_	com	fr_	org
1	co_	com	de_	ee_	fr_	net	org	uk_

- (b) Sortieren Sie nun die gleiche Reihung wieder schrittweise, diesmal jedoch unter Verwendung des Mergesort-Verfahrens (Sortieren durch Mischen). Erstellen Sie dazu eine Tabelle wie das folgende Muster und vermerken Sie in der ersten Spalte jeweils welche Operation durchgeführt wurde: Wenn Sie die Reihung geteilt haben, schreiben Sie in die linke Spalte ein T und markieren Sie die Stelle, an der Sie die Reihung geteilt haben, mit einem senkrechten Strich „|“. Wenn Sie zwei Teilreihungen durch Mischen zusammengeführt haben, schreiben Sie ein M in die linke Spalte und unterstreichen Sie die zusammengemischten Einträge. Beginnen Sie mit dem rekursiven Abstieg immer in der linken Hälfte einer (Teil-)Reihung.

```

0 | Reihung
T | de_   com   uk_   org | co_   net   fr_   ee_
T | de_   com | uk_   org
T | de_ | com
M | com   de_

```

T				uk_	org				
M				org	uk_				
M	com		de_	org	uk_				
T						co_	net		fr_ ee_
T						co_	net		
M						co_	net		
T								fr_	ee_
T								ee_	fr_
M						co_	ee_	fr_	net
M	co_	com	de_	ee_	fr_	net	org	uk_	

- (c) Implementieren Sie das Sortierverfahren Quicksort für String-TLDs in einer gängigen Programmiersprache Ihrer Wahl. Ihr Programm (Ihre Methode) wird mit drei Parametern gestartet: dem String-Array mit den zu sortierenden TLDs selbst sowie jeweils der Position des ersten und des letzten zu sortierenden Eintrags im Array.

Lösungsvorschlag

```

public class Quicksort {

    public static void swap(String[] array, int index1, int index2) {
        String tmp = array[index1];
        array[index1] = array[index2];
        array[index2] = tmp;
    }

    public static int partition(String[] array, int first, int last) {
        int pivotIndex = (last + first) / 2;
        String pivotValue = array[pivotIndex];
        int pivotIndexFinal = first;
        swap(array, pivotIndex, last);
        for (int i = first; i < last; i++) {
            if (array[i].compareTo(pivotValue) < 0) {
                swap(array, i, pivotIndexFinal);
                pivotIndexFinal++;
            }
        }
        swap(array, last, pivotIndexFinal);
        return pivotIndexFinal;
    }

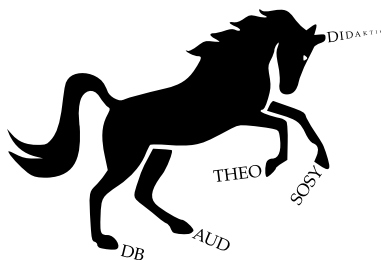
    public static void sort(String[] array, int first, int last) {
        if (first < last) {
            int pivotIndex = partition(array, first, last);
            sort(array, first, pivotIndex - 1);
            sort(array, pivotIndex + 1, last);
        }
    }

    public static void main(String[] args) {
        String[] array = new String[] { "de", "com", "uk", "org", "co", "net", "fr",
        ↪ "ee" };
        sort(array, 0, array.length - 1);
    }
}

```

```
for (int i = 0; i < array.length; i++) {  
    System.out.println(array[i]);  
}  
}  
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen\\_66115/jahr\\_2017/fruehjahr/Quicksort.java](https://github.com/bschlangaul/examen/examen_66115/jahr_2017/fruehjahr/Quicksort.java)



## Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an [hermine.bschlangaul@gmx.net](mailto:hermine.bschlangaul@gmx.net). Der TeX-Quelltext dieser Aufgabe kann unter folgender URL aufgerufen werden: <https://github.com/bschlangaul-sammlung/examens-aufgaben-tex/blob/main/Examen/66115/2017/03/Thema-1/Aufgabe-2.tex>