

Radixsort

Weiterführende Literatur:

- Algorithmen und Datenstrukturen: Tafelübung 11, WS 2018/19, Seite 68-75
- Wikipedia-Artikel „Radixsort“

RadixSort ; verallgemeinertes Sortieren durch Fachverteilen • Voraussetzung: • Elemente sind Zeichenfolgen über einem endlichen Alphabet Σ • auf den Zeichen des Alphabets ist eine Totalordnung definiert • für jedes Zeichen des Alphabets steht ein Fach zur Verfügung • einzelne Stellen aller Elemente von hinten nach vorne abarbeiten: Partitionieren ; Elemente abhängig von der aktuellen Stelle in das passende Fach legen Einsammeln ; Elemente wieder aus den Fächern rausnehmen • vom Fach für das „kleinste“ Zeichen zum Fach für das „größte“ Zeichen • relative Reihenfolge der Elemente in einem Fach muss erhalten bleiben!

```
3 import java.util.*;
4
5 /**
6  * <a href="https://www.geeksforgeeks.org/radix-sort/">geeksforgeeks.org</a>
7  */
8 public class Radix extends Sortieralgorithmus {
9     // A utility function to get maximum value in arr[]
10    static int getMax(int arr[], int n) {
11        int mx = arr[0];
12        for (int i = 1; i < n; i++)
13            if (arr[i] > mx)
14                mx = arr[i];
15        return mx;
16    }
17
18    // A function to do counting sort of arr[] according to
19    // the digit represented by exp.
20    void countSort(int arr[], int n, int exp) {
21        int output[] = new int[n]; // output array
22        int i;
23        int count[] = new int[10];
24        Arrays.fill(count, 0);
25
26        // Store count of occurrences in count[]
27        for (i = 0; i < n; i++)
28            count[(arr[i] / exp) % 10]++;
29
30        // Change count[i] so that count[i] now contains
31        // actual position of this digit in output[]
32        for (i = 1; i < 10; i++)
33            count[i] += count[i - 1];
34
35        // Build the output array
36        for (i = n - 1; i >= 0; i--) {
37            output[count[(arr[i] / exp) % 10] - 1] = arr[i];
38            count[(arr[i] / exp) % 10]--;
39        }
40
41        // Copy the output array to arr[], so that arr[] now
42        // contains sorted numbers according to current digit
```

```

43     for (i = 0; i < n; i++)
44         arr[i] = output[i];
45 }
46
47 // The main function to that sorts arr[] of size n using
48 // Radix Sort
49 public int[] sortiere() {
50     // Find the maximum number to know number of digits
51     int m = getMax(zahlen, zahlen.length);
52
53     // Do counting sort for every digit. Note that
54     // instead of passing digit number, exp is passed.
55     // exp is 10^i where i is current digit number
56     for (int exp = 1; m / exp > 0; exp *= 10)
57         countSort(zahlen, zahlen.length, exp);
58     return zahlen;
59 }
60
61 public static void main(String[] args) {
62     new Radix().teste();
63 }
64 }

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bachelorlangaul/sortier/Radix.java](https://github.com/src/main/java/org/bachelorlangaul/sortier/Radix.java)

Literatur

- [1] *Algorithmen und Datenstrukturen: Tafelübung 11, WS 2018/19.* https://www.studon.fau.de/file2567217_download.html. FAU: Lehrstuhl für Informatik 2 (Programmiersysteme).
- [2] *Wikipedia-Artikel „Radixsort“.* <https://de.wikipedia.org/wiki/Radixsort>.