**Quick Sort (Schnelles Sortieren)**

Der Quick Sort-Algorithmus ist ein Sortierverfahren, das durch rekursives Einteilen der Elemente in Bezug zur Größe eines Pivot-Elements funktioniert. Das Verfahren wird z.B. hier erklärt: <https://youtu.be/UoJJ78K-uc0>

**Aufgabe 1: Beschreibung des Verfahrens**

Ergänzen Sie die Beschreibungen zu dem Quick Sort-Algorithmus.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| unsortiert | | | | 1.  2.  3.  4. |
|  | | | |
| P | unsortiert | | |
|  | | | |
| unsortiert und kleiner als P | | P | unsortiert und größer als P |
|  | | | |
| sortiert und kleiner als P | | P | sortiert und größer als P |
|  | | | |
| sortiert | | | |

**Aufgabe 2: Sortieren**

a) Geben Sie die fehlenden Schritte der Sortierung für die Liste zahlen=[5,7,2,4,6,9,3,1,8] an.

|  |  |  |
| --- | --- | --- |
| **Schritt** | **Listen** | **Listen** |
| / | [5,7,2,4,6,9,3,1,8] |  |
| 1 | [5,7,2,4,6,9,3,1,8] |  |
| 2 | [2,4,3,1][5][7, 6, 9, 8] |  |
| 1 |  |  |
| 2 |  |  |
| 2, 3 |  |  |
| 4 |  |  |
| 4 | [1,2,3,4][5][6,7,8,9] |  |
| 4 |  |  |

b) Führen Sie das Sortierverfahren für die Liste zahlen=[13,71,6,1,15,81,100,66,7,12] durch.

**Aufgabe 3: Der Quick Sort-Algorithmus**

a) Vervollständigen Sie die funktionale Implementierung des Quick Sort-Algorithmus mit Hilfe der folgenden Ausdrücke:

|  |  |  |  |
| --- | --- | --- | --- |
| quicksort | [] | ++ | -> |
| [Int] | quicksort | ++ | [Int] |
| [x | x <- es, x > e] | | [x | x <- es, x <= e] | |

|  |
| --- |
| quicksort ::  quicksort [] =  quicksort (e:es) = |

b) Reduzieren Sie Sie den Aufruf der Funktion mit der Eingabeliste [24, 7, 36, 5, 3, 12, 48, 25, 27].

|  |
| --- |
| quicksort [24, 7, 36, 5, 3, 12, 48, 25, 27]  🡪  🡪  🡪  🡪 |

b) Recherchieren Sie nach einer imperativen Implementierung des Algorithmus und vergleichen Sie beide Varianten (z.B. hier: <https://javabeginners.de/Algorithmen/Sortieralgorithmen/Quicksort.php>).

|  |
| --- |
|  |

**Aufgabe 4: Effizienzbetrachtung**

Geben Sie eine Eingabeliste an, für die das Sortierverfahren besonders effizient arbeitet, sowie eine Eingabeliste, die zu einer ineffizienten Abarbeitung führt.

|  |  |  |
| --- | --- | --- |
|  | **Effiziente Abarbeitung** | **Ineffiziente Abarbeitung** |
| **Beispiel** |  |  |
| **Begründung** |  |  |