

Quicksort

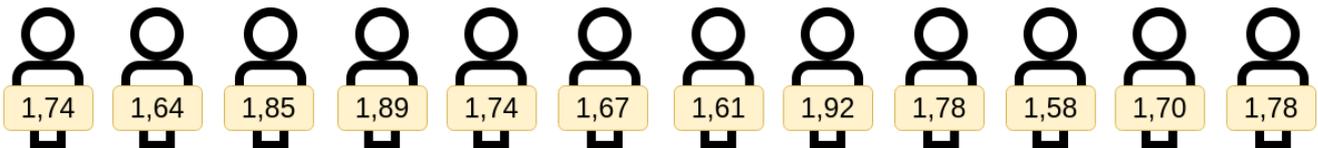


Um den Quicksort Algorithmus verstehen und implementieren zu können, sollte man die Abschnitte [Rekursion](#) und das [Teile-und-Herrsche-Prinzip](#) bearbeitet haben.

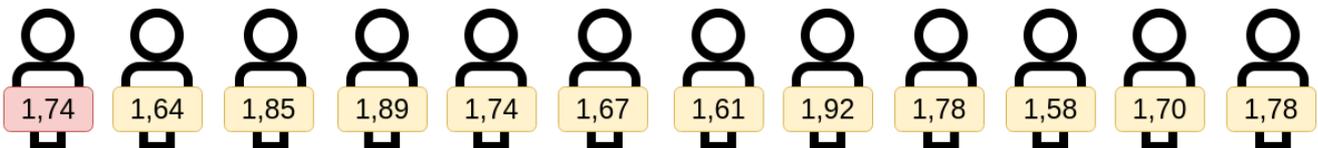
Quicksort ist ein sehr schneller Sortieralgorithmus. Er kommt in der Praxis häufig zum Einsatz. Zahlreiche Standardbibliotheken verschiedener Programmiersprachen enthalten Methoden um zum Beispiel Arrays zu sortieren, die in als Quicksort implementiert sind. Zum Beispiel hat die Standardbibliothek der Programmiersprache C eine Funktion namens `qsort`. Quicksort verwendet ein [Teile-und-herrsche-Prinzip](#).

Modellvorstellung

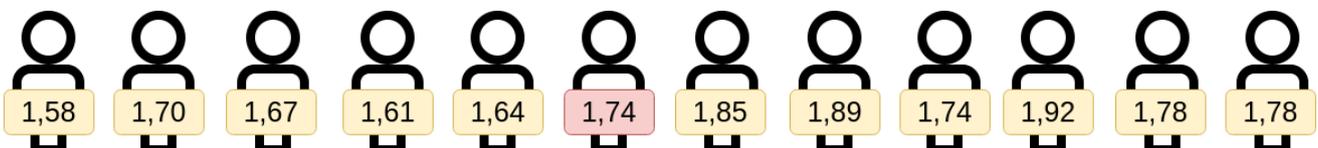
Stell dir vor die Schüler der 7a wollen sich wie die Orgelpfeifen der Größe nach geordnet aufstellen:



Zunächst wählt man die erste Person als "Vergleichsgröße" aus, der Fachbegriff für das Element, das als Vergleichselement verwendet wird ist **Pivotelement**.



Jetzt teilt man das Problem in zwei Unterprobleme auf: Alle Schülerinnen die kleiner als das Pivotelement sind stellen sich links davon auf, alle die größer oder gleich sind rechts:



Das Pivotelement scheidet jetzt aus dem Verfahren aus, es bleibt an dem Platz, an dem es sich jetzt befindet. Jetzt haben wir zwei "kleinere" Sortierprobleme geschaffen: Die Schülerinnen, die links stehen sind ungeordnet, die größeren auf der rechten Seite ebenfalls. Auf jeden Fall – auch im ungünstigsten¹⁾ – sind die neuen Teilproblem(e) kleiner als das ursprüngliche Problem.

¹⁾

welches ist der ungünstigste Fall?

From:
<https://info-bw.de/> -

Permanent link:
<https://info-bw.de/faecher:informatik:oberstufe:algorithmen:sortieren:quicksort:start?rev=1643290599>

Last update: **27.01.2022 13:36**

