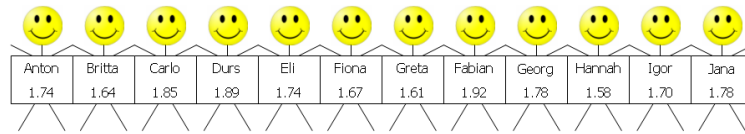


Sortieren durch Zerlegen / Quicksort

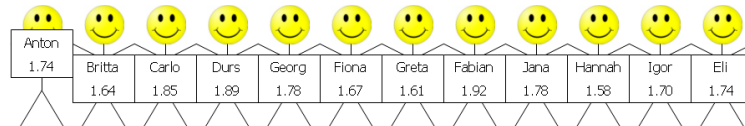
Ein Sortierspiel:

Anton, Britta, Carlo, ... wollen sich der Größe nach in einer Reihe aufstellen. Zuerst werden alle genau vermessen.

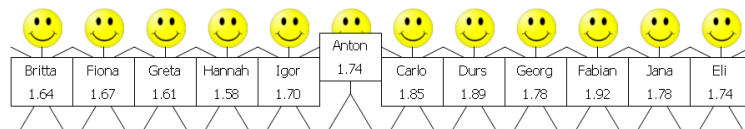


Das Spiel läuft jetzt so ab:

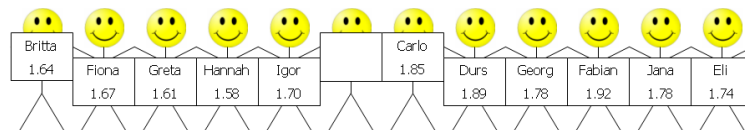
Eine Person (z.B. die erste in der Reihe) wird als Vergleichsperson ausgewählt. Im vorliegenden Beispiel ist das Anton.



Alle anderen ordnen sich links bzw. rechts von der Vergleichsperson ein, je nachdem, ob sie kleiner oder größer/gleich als die Vergleichsperson sind.



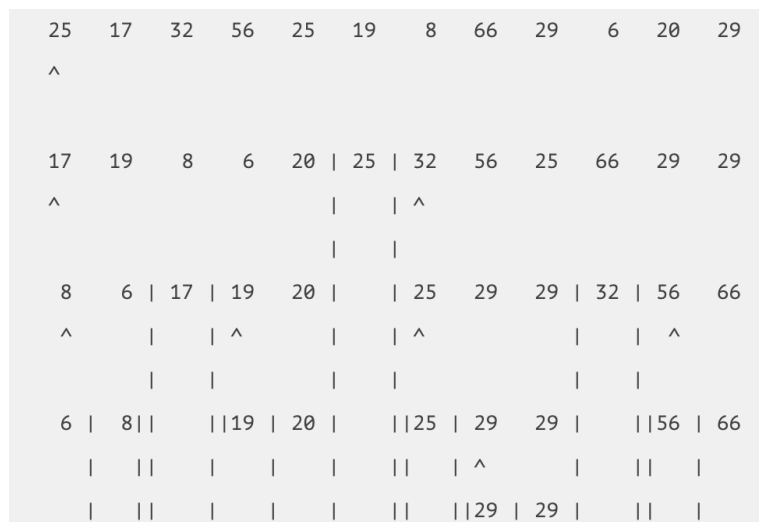
Anton bleibt jetzt auf seiner Position. Er nimmt nicht mehr an weiteren Spielrunden teil. Dasselbe Spiel wird jetzt im linken und im rechten Bereich durchgeführt: Eine Person (z.B. die erste in der Reihe) wird wiederum als Vergleichsperson ausgewählt.



Und so weiter ...

Die Grundidee

Gegeben ist eine Liste mit vergleichbaren Elementen. Ein Element der Liste (z.B. das erste Element) wird ausgewählt. Die Restliste wird dann gemäß dieses Elements in zwei Teillisten aufgeteilt: die Liste der Elemente der Restliste, die kleiner als das ausgewählte Element sind sowie die Liste der Elemente der Restliste, die nicht kleiner (d.h. größer oder gleich) als das ausgewählte Element sind. Dieser Vorgang wird mit den Teillisten völlig analog fortgesetzt.





Führe das S

[illegible]

Aufgabe 3

Implementiere den Algorithmus **quicksort** mithilfe der untenstehenden Ablaufmodellierung.

Ablaufmodellierung mit mehreren Listen

Das oben erläuterte Sortierverfahren soll jetzt mit einem Algorithmus beschrieben werden. Informell lässt sich dieser Sortierablauf so beschreiben:

ALGORITHMUS

def quicksort (L)

Übergabeparameter der Funktion: Liste L

Funktionskörper:

wenn die Liste L mehr als ein Element hat:

 # zerlegen

 wähle als Pivotelement¹ p das erste Element der Liste aus

 erzeuge Teillisten K (kleiner) und G (größer) aus der Restliste (L ohne p) mit:

 - alle Elemente aus K sind kleiner als das Pivotelement p

 - alle Elemente aus G sind größer gleich als das Pivotelement p

 # quicksort auf die verkleinerten Listen anwenden

 KSortiert = quicksort(K)

 GSortiert = quicksort(G)

 # zusammensetzen

 LSortiert = KSortiert + [p] + GSortiert

sonst:

 LSortiert = L

Rückgabe: LSortiert

¹Pivotelement = das erste von einem Algorithmus ausgewählte Element einer Zahlenmenge