Quicksort (schnelles Sortieren)

Die Grundidee ist, dass beim Quicksort-Algorithmus immer ein Vergleichselement herausgesucht wird und mit allen anderen Elementen verglichen wird. Nun gibt es eine Liste, deren Elemente alle kleiner und eine Liste deren Elemente alle größer als das Vergleichselement sind. Mit diesen beiden Teillisten wird wieder so verfahren, bis alle Elemente sortiert sind.

Beschreibung des Algorithmus:

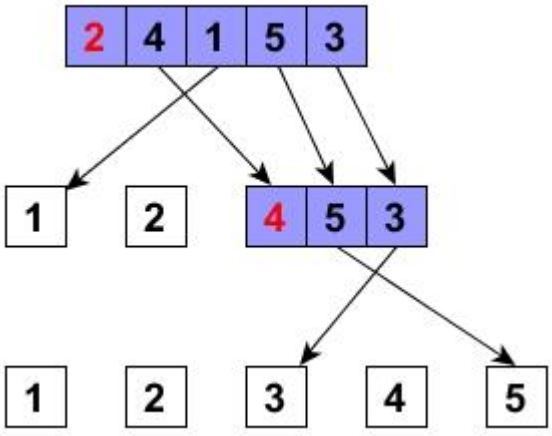
Das erste Element der Liste ist das Pivotelement (Vergleichselement). Nun wird jedes Element der restlichen Liste nacheinander mit diesem verglichen. Jedes kleinere (leichtere) Element wird in die „linke“ Teilliste eingefügt und jedes größere (schwerere) Element wird in die „rechte“ Teilliste eingefügt. Das Ergebnis ist, dass man nun eine Teilliste mit nur kleineren (leichteren) Elementen und eine Teilliste mit nur größeren (schwereren) Elementen als das Pivotelement besitzt. Daher ist das Pivotelement zwischen den beiden Teillisten in der gesamten Liste schon richtig einsortiert.

Mit den beiden Teillisten wird nun getrennt voneinander genauso verfahren. Das heißt, dass wieder das erste Element das Pivotelement wird und alle anderen (dieser Teilliste) mit diesem verglichen werden.

Teillisten, die nur noch ein Element enthalten sind logischerweise sortiert und markieren somit den Abbruchsfall.

Beispiel:

Die roten Elemente kennzeichnen jeweils das Pivotelement für die jeweilige Teilliste.



Aufgaben

1.) Lese die Anleitung samt Beispiel zu dem gegebenen Algorithmus genau durch. Hole einen Materialiensatz vom Lehrerpult.

2.) Sortiere die Cornflakes-Packungen mithilfe des gegebenen Algorithmus aufsteigend nach Gewicht.

Die Startreihenfolge dafür ist: Cornflakes 1, Honey Bsss Pops, Cornflakes 2, Choco Krispies Chocos, Frosties, Smacks, Choco Krispies, Special K Classic.

Notiere zudem, wie häufig mit diesem Algorithmus gewogen werden muss, bis die Cornflakes-Packungen nach Gewicht sortiert sind.

3.) Schätze anhand der Anzahl der zu vergleichenden Elemente im Vergleich zu der Anzahl der Wiegungen ab, in welcher Laufzeitkategorie der Algorithmus liegen müsste. (Bedenke, dass der Computer für jede Wiegung mehr als einen Rechenschritt benötigt!)

4.) Begründe in Worten anhand der Beschreibung des Algorithmus, warum deine Abschätzung der Laufzeitkategorie stimmt.

Zusatzaufgaben:

Z1) Begründe, warum es zulässig ist, nur die Anzahl der Wiegungen zur Abschätzung des Laufzeitaufwands heranzuziehen.

Z2) Überlege, wie die Startreihenfolge angeordnet sein muss, so dass der Algorithmus erst nach möglichst vielen Schritten beendet ist.

Bestimme, in welcher Laufzeitkategorie der Algorithmus mit dieser (speziellen) Startreihenfolge liegen würde.