

Sortieralgorithmen am Gymnasium

Studentin: *Alexandra Maximova*

Fach: *Fachdidaktik 2 (Die Erfolgsgeschichte von Computer Science Unplugged)* – Dozent:
Giovanni Serafini
Abgabedatum: 06.05.2020

Zielsetzung

Sortieren wird im Alltag immer mehr vom Rechner übernommen und durch die Suche ersetzt. Dort, wo man früher ein Wörterbuch durchblättern musste und sehr froh war, dass die Wörter darin alphabetisch sortiert sind, kann man heute das gesuchte Wort in ein Suchfeld eingeben und direkt den Eintrag lesen. Früher musste man die eigene Kontaktliste (handgeschrieben auf Papier) durchlesen, um Adressen und Telefonnummern von Freunden zu finden, und war froh, wenn sie einigermaßen nach dem ersten Buchstaben vom Namen sortiert waren. Heute die Kontaktliste im eigenen Handy ist immer alphabetisch sortiert, und zusätzlich hat man die Möglichkeit, nach dem Namen zu suchen.

Sortieren und Umsortieren geht blitzschnell, man muss nur auf das richtige Button drücken, und im Nu sind die Einträge nach Datum sortiert; Noch ein Klick – und sie sind nach Name sortiert. Man hat den Eindruck, dass es automatisch passiert und überhaupt keinen Aufwand braucht.

In diesem Kontext ist es schwierig den SuS zu erklären, warum man überhaupt etwas zu sortieren braucht, wenn man Ctrl+F benutzen kann und viel schneller etwas finden, als "manuell" in einer alphabetisch sortierten Vokabelliste.

Ein Ziel dieser Lerneinheit könnte sein, den SuS zu zeigen, was hinter einem einzigen Klick stecken kann. Der Computer wird also aus einem einfachen schwarzen Kasten zu einem bunten Universum von faszinierenden Prozessen und Algorithmen.

Noch wichtiger scheint mir, den SuS das Konzept der **Laufzeit** näher zu bringen. Sortieralgorithmen eignen sich hervorragend für eine erste Begegnung damit, weil es mehrere einfach zu verstehende Algorithmen gibt, die genau das gleiche machen, aber eine unterschiedliche Laufzeit haben.

Auf die Einführung der Landau-Notation würde ich in diesem Rahmen verzichten. Die SuS sollten lernen, wie man Algorithmen bezüglich ihrer Laufzeit vergleichen kann, nämlich indem man die **Anzahl der Vergleiche** zählt, die man braucht, um eine Liste zu sortieren, und wie sich diese Anzahl in Abhängigkeit der **Länge der Liste** verändert. Eine andere interessante Ebene, die Verfahren wie Quicksort oder Bubblesort zur Untersuchung anbieten, ist der Unterschied zwischen **Worst Case** und **Best Case**.

Voraussetzungen

Gymnasium, zweites Semester des ersten Jahres oder erstes Semester des zweiten Jahres. Grundlagefach Informatik. Richtung Mathe-Physik. Programmieren müssen sie

nicht unbedingt können. Rekursion müssen sie schon kennen.

Sie können nach Anleitung arbeiten und selber präzise genug Anweisungen geben, so dass andere Leute sie folgen können.

Arbeitsschritte

Ich würde es mega schön finden, wenn sie selber Verfahren entwickeln könnten. Jede Gruppe von 4 Personen kriegt die Waage und 8 Behälter (in Wirklichkeit 16, aber am Anfang benutzen sie nur 8). Sie müssen einen Sortierverfahren möglichst genau beschreiben, so dass eine andere Gruppe nachvollziehen kann, was passiert. Dann geben sie tatsächlich das eigene Algorithmus einer anderen Gruppe. Die andere Gruppe muss es ausführen können und Vergleiche zählen. Dann genau dasselbe mit 16 Behälter machen und sehen, wie sich die Anzahl Vergleiche verändert.

Ok, geht vielleicht nicht mega gut. Ich glaube, wir sollten nach der ersten Runde zusammentragen, und ich soll sicher stellen, dass zumindest Selection Sort und Merge Sort rauskommen. Wenn die SuS nicht drauf gekommen sind, verteile ich Arbeitsblätter mit diesen neuen Verfahren, die sie miteinander vergleichen müssen. Was ist besser? Was ist schneller? Was passiert, wenn wir die Länge der Liste verdoppeln? Wie viele Vergleiche brauchen wir in Abhängigkeit von der Länge der Liste (als Funktion)?

Etwas anderes, was mir im Arbeitsblatt von Jens Gallenbacher sehr gefallen hat, sind die Schätzungen. Bevor sie das mit 16 Behälter tatsächlich ausführen, müssen sie schätzen, wie viele Vergleiche sie dafür brauchen werden. Und dann schauen, wie weit das tatsächliche von dem geschätzten ist.

Bezug zur Allgemeinbildung

So haben sie ein Instrument, um Algorithmen miteinander zu vergleichen. Und sie verstehen, dass das Computer sehr schnell ist, aber nicht unendlich schnell, und für sehr grosse Datenmengen braucht er doch sichtbare Zeit, um etwas zu sortieren.

Eignung Altersstufe

In diesem Alter können sie Funktionen aufstellen, um die Zusammenhänge besser zu quantifizieren. Deswegen sollten sie das auch tun. Gruppenarbeiten verstärken Kommunikationsfähigkeiten und Arbeitsaufteilung.

Die Aktivität mit greifbaren Gegenständen macht die Geistesbits im Computerinneren weniger abstrakt. Ich glaube, vieles versteht man besser durch ausprobieren.