

SORTIEREN im Informatikunterricht

Dieses Thema unterrichte ich gerne, da dabei der Aspekt des 'Tüftelns' nicht zu kurz kommt, der sonst im Informatikunterricht oft fehlt. Programmieren hat viel mit dem Lösen von Denksportaufgaben zu tun, wenn es um Algorithmen geht. Und das macht einfach Spaß.

Wer die Beispiele durchsieht wird feststellen, dass Python sowohl den Logo-Zugang mit Rekursion und Listen, als auch den sequentiellen Zugang der imperativen Sprachen zwanglos ermöglicht (und wer will – es geht auch funktional). Sprachen wie Delphi oder C++ zwingen den Schüler erst einmal, 'nur nichts falsch' zu machen. Das mag Sprach-Juristen erfreuen, lässt aber den Forschungsdrang der Schüler verkümmern.

Einstiegsaufgaben:

- 1.) Du hast **zwei Variablen** a und b. Schreib einige Programmzeilen, sodass die beiden ihren Wert tauschen. Dann schreib ein paar Zeilen, damit a die kleinere Zahl enthält.

Hier wird in C oder Pascal von den Schülern zuerst meist die Idee $a=b$ und $b=a$ geboren, aber rasch verworfen. Dann folgt der bekannte Dreiertausch $x=a$; $a=b$; $b=x$.

Was kann Python, was viele andere nicht können? Die Mehrfachzuweisung $a,b=b,a$.

Man kommt dann zu

```
def test(a,b):  
    print 'vorher :', a,b  
    if a>b: a,b=b,a  
    print 'nachher:', a,b
```

und man kann mit `test(3,5)`, `test(4,2)` sofort alles ausprobieren. Aber auch – und das überrascht wohl alle, die Python noch nicht kennen - `test('eins','acht')` klappt mit dem selben Progrämmchen.

- 2.) Du hast **drei Variablen** a, b und c. Schreib einige Programmzeilen, sodass nachher die Werte $a \leq b \leq c$ aufsteigend sortiert sind.

Nun ist meist Tüfteln angesagt. Bis jetzt entstand aber immer eine brauchbare Lösung. Schüler miteinander diskutieren lassen!

- 3.) Du hast **eine Liste mit 3 Werten**, etwa `[3,2,5]` oder `['Eric','Michael','John']`. Schreib einige Programmzeilen, sodass die Werte in der Liste der Größe nach aufsteigend sortiert sind.

Hier kann man den Zugriff auf Listenelemente und die Slices wiederholen. Meist ist die Lösung eine Adaption von 2.)

- 4.) nun kann man den Schülern eine Liste aus 4, 5 oder 6 Elementen androhen.

5.) und darauf hinweisen, dass sortieren eine der Hauptbeschäftigungen eines PC ist. Schließlich legt man ja auch Wert auf eine geordnete Anzeige des Inhaltsverzeichnisses im Explorer. Und wie berechnet die Rendering-Engine des neuesten Ballerspiels wohl die Sichtbarkeit von Gegenständen? Und wie teilt der Prozessor Aufgaben verschiedener Priorität die Rechenzeit zu? Und wie reagiert ein Aufzug, wenn in mehreren Stockwerken Leute einsteigen wollen?

- 6.) und schon hat der Lehrer einige Sortierideen parat, die auch für ganz große Listen klappen.

7.) Fächerübergreif: Wie sortiert man Worte? Wie später angedeutet – es gibt gar keine allgemein gültige

Regel. Ob man wie in neuen Lexika ä=a setzt oder wie in alten Lexika ä=ae setzt, oder eine absurde Lösung wie im Österr. Telefonbuch ('Ärzte' hinter 'Azteke') – hier gibt es viel zu hinterfragen (ist ein scharfes ß ein ss oder sz? Der Physiklehrer kennt die einzige Ausnahme. Oder ein eigenes Zeichen? Wenn ja – wo im Alphabet reihen wir es ein?). Und die Telefonbuchlösung hat sogar mit Informatik zu tun: Wenn Umlaute zu Doppelzeichen werden, muss der einzusortierende Text während der Behandlung umcodiert werden. Von Adelsprädikaten ausländischer Schriftsteller weiß der Schulbibliothekar ein klagendes Lied zu singen.

Warnung:

Ausgefuchste Minimierung der Laufzeit wird in den Beispielprogrammen NICHT betrieben. Sie ist nämlich stark programmiersprachen-, prozessor- und hardwareabhängig, und für unterschiedliche Datentypen müssen unterschiedliche Lösungen gefunden werden. Wenn man das Prinzip (den Algorithmus) jedoch wirklich verstanden hat, ist eine Umsetzung auf spezielle Anforderungen kein Kunststück mehr. Und diese Grundlage sollte der Unterricht bieten.

Erweiterungen:

- Wie sortiert man riesige Felder? Die Methoden sind nicht allzu kompliziert und für Interessierte eine nette Anwendung. (Externe Methoden siehe z.B. Sedgewick)
- Was sind 'linked lists' und wofür verwendet man sie beim Sortieren?
- Animation der Routinen: Wenn man etwa für Selection Sort die Swap-Funktion erweitert, sodass sie bei jedem Aufruf die aktuelle Liste anzeigt, kann man 'zusehen', wie die Elemente verschoben werden. Dies kann man auch in Grafik darstellen.
- Es gibt noch weitere 'Standardmethoden' wie Comb-Sort, Bucket-Sort, Radix Exchange und Straight Radix Sort, Tree Sorts, Sample Sort, Distribution Count, etc. Ideales Futter für Schüler/Kurse mit Forscherdrang.

Literatur:

Wer mir nicht glaubt, mehr Details braucht oder einfach in einem guten Buch blättern will: folgende gewichtige Werke zum Thema sind ehrlich empfehlenswert (was bedeutet, dass ich selbst sie gern konsultiere):

1.) die Bibel der Algorithmiker:

Donald **Knuth**: The Art of Computer Programming. Teil 3 – Sorting and Searching.

Immense Fülle von Informationen, detaillierte Angaben zur Umsetzung, unendlich viele weiterführende Ideen und schräge Einfälle. Aber nicht ganz leicht....

2.) die Praxis:

Press,...: Numerical Recipies in C

Eigentlich ist es ein Buch über numerische Methoden, aber ein ausgezeichnetes Kapitel übers Sortieren ist dabei.

3.) das Lehrbuch:

Sedgewick: Algorithms

auch auf deutsch erhältlich und in verschiedenen Geschmacksrichtungen (Pascal, C, Cplusplus oder C++ Syntax der Beispielprogramme). Solide Grundlagen, detaillierte Erklärungen, anschauliche Beispiele.

Einzigster Nachteil: alle bevorzugen (aus Performance-Gründen) die iterativen Lösungen, die dem menschlichen Gedankengang nicht so ganz nahe liegen. Ich habe bei meinen Beispielen versucht, wenn sinnvoll, beide Gesichtspunkte zu erwähnen.