**unplugged**

„Computer Science unplugged“ ist ein von der Universität Canterbury (Neuseeland) entwickeltes Format, bei Konzepte der Informatik ohne Computer („unplugged“) nacherlebt und gelernt werden können. Das vorliegende Material orientiert sich an diesem Ansatz, steht aber in keiner Verbindung zu dieser Universität.

Suchen: eine wiederkehrende Aufgabe

Du kennst das Problem: Eine Suchmaschine hat Dich auf eine Seite verwiesen, auf der Dein Suchbegriff vorkommt - irgendwo in den 5-6 Bildschirmseiten Text. Du drückst Strg-F (oder, am Smartphone, die entsprechenden Buttons für „auf der Seite suchen“), gibst Deinen Suchbegriff ein und schon siehst Du, wo der Begriff zum ersten Mal verwendet wird.

Doch was passiert hier eigentlich?



Das menschliche Vorgehen in Worte fassen

- ① Versuche einmal, Dein Vorgehen in Worte zu fassen, wenn Du in der Definition für „unplugged“ nach dem Wort „Material“ suchst und schreibe das Ergebnis auf.
- ② Tausche Deine Notizen mit einem*r Mitschüler*in aus und versuche es mit einer fremden Anleitung - seid ihr beide gleich vorgegangen?
- ③ Besprecht, welche Fähigkeiten für euer Vorgehen benötigt werden, und ob ein Computer in der Lage ist, diese zur Verfügung zu stellen.
- ④ Besprecht, wie sich die Suchzeit ändert, wenn der Text doppelt, halb, viermal,... so lang wäre. Das gesuchte Wort sei zufällig gewählt, kann also an jeder Position des Textes stehen.

...und wie lange dauert das nun?

Zur Messung des Zeitbedarfs eignen sich Sekunden, Millisekunden, Minuten usw. sehr schlecht: Je nach dem auf welchem System mit welchen Spezifikationen die selbe Suche durchgeführt wird. In der Informatik verwendet man stattdessen üblicherweise die Anzahl der „ist gleich“-Operationen (also der Vergleiche).

Außerdem unterscheidet man einen Optimalfall (Best Case, bei einer Suche also „gefunden beim ersten Versuch“), einen schlechtesten Fall (worst case - „gefunden, nachdem alle anderen Elemente erfolglos verglichen wurden“) und einen durchschnittlichen Fall (average case - „so lange dauert es im Durchschnitt, bis ein Element gefunden wird“)

- ⑤ Besprecht, wie viele Vergleiche für das Suchen eines Wortes im Text ganz oben nach eurer Methode im best case und worst case notwendig sind, und wie sich diese Zahlen ändern, wenn der Text doppelt oder viermal so lang ist.
- ⑥ Ergeben sich Möglichkeiten, Vergleiche einzusparen, wenn die Wörter alphabetisch sortiert wären? Diskutiert.

Einfacher Suche: vom Anfang bis zur Fundstelle (linear)

Eine Lösung, die viele von euch gefunden haben, besteht darin, beim ersten Wort anzufangen, und dann Wort für Wort zu vergleichen, ob es sich das gesuchte Wort handelt. So arbeitet man sich durch den Text, bis man auf das Wort trifft.

Der best case ist dabei unabhängig von der Textlänge - wenn man direkt beim ersten Versuch das Wort findet, braucht man immer nur einen Vergleich. Im worst case ist das allerletzte Wort des Textes gesucht. Dann muss man für einen doppelt so langen Text auch doppelt so viele Vergleiche machen, bis man es gefunden hat.

Dieses Suchverfahren wird „lineare Suche“ genannt, weil der Text wie beim Lesen Wort für Wort (eben: linear) durchgegangen wird.



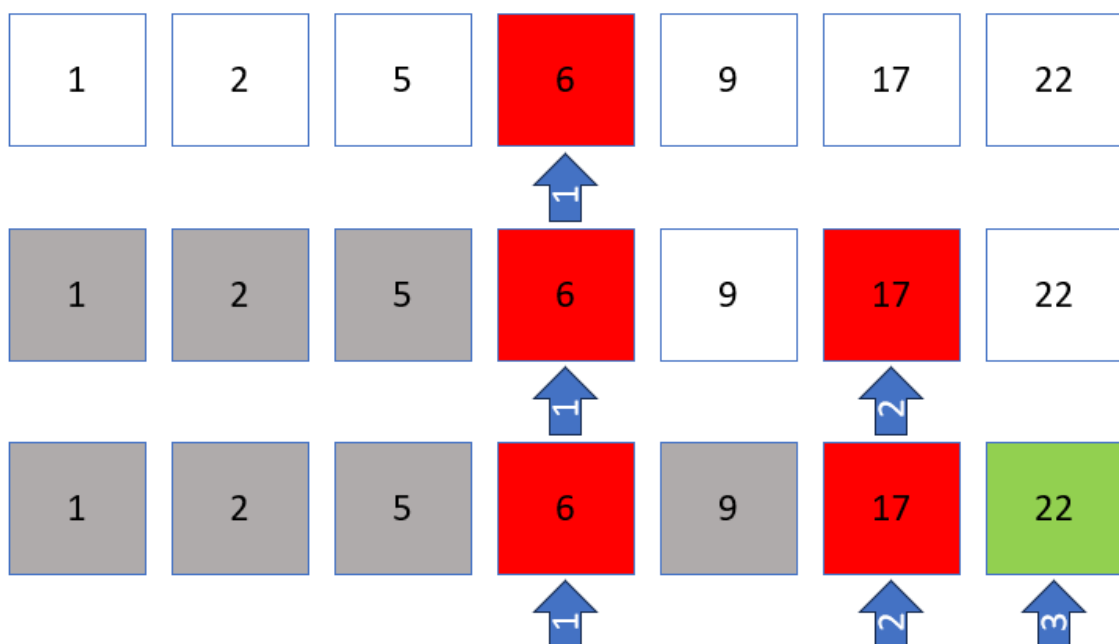
Übrigens: Was durchsucht wird, ist völlig egal. Wir können Texte, Listen, Bilder, Bilderschriften, ... durchsuchen, solange wir nur eine Möglichkeit des Vergleichens haben.

Optimierung bei vorsortierten Listen

Hat man eine bereits aufsteigend oder absteigend sortierte Liste vor sich (das ist bei Texten normalerweise nicht der Fall!), kann man sich besonders bei längeren Texten Suchzeit in Form von Vergleichen sparen. Dabei macht man sich die Information zunutze, dass die Elemente bereits sortiert sind.

Beispiel: Gesucht wird die 22. Mit der linearen Suche wären es im worst case 7 Vergleiche.

⑦ Beschreibe, wie die Vorsortierung der Liste im folgenden Verfahren genutzt wird!



⑧ Wie verändert sich die Anzahl der Vergleiche, wenn die Liste doppelt so lang ist?

