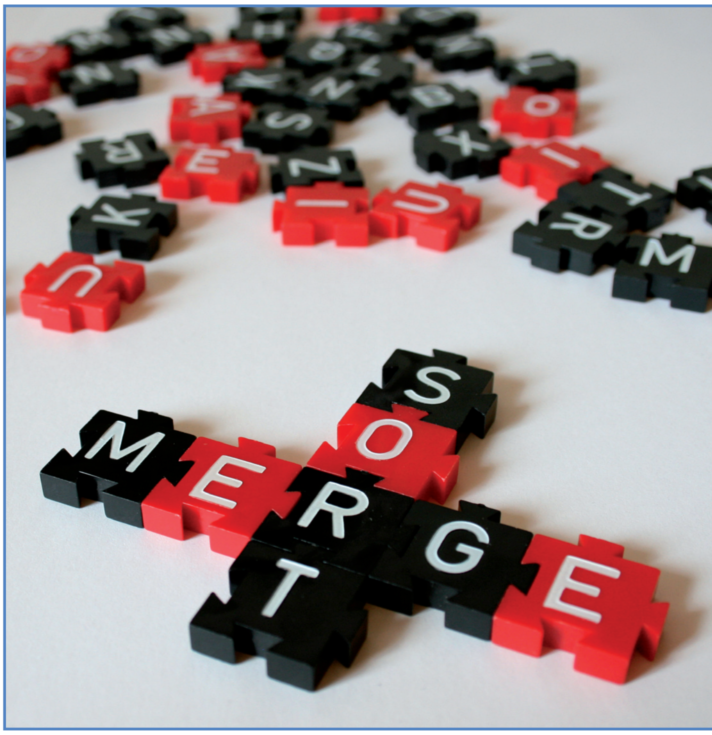
Modul So4 Mergesort



Zeitrahmen
40 Minuten

Zielgruppe
Sekundarstufe I

Lehrziel
Vermittlung eines effizienten, relativ leicht durchschaubaren Sortierverfahrens mit Ausblicken auf Rekursion und rekursive Datenstrukturen (natürlicher Binärbaum).

Requisiten
Geburtstagsordnung: keine

Partizipanden
Experimentalgruppe: n ≥ 10 Jugendliche
Algoritm. Unterstützung: 1 Aufteiler (stellt jeweils die Mitte fest und sichert, dass sich beide Hälften solange dies möglich ist, weiter teilen)
1 Vergleicher (Steuert die Mischoperation)

Beobachtungs(gruppe):
Mischzähler: Zähler der aufzeichnet, wie oft eine Person im Zuge der Misch- und Rücksetzoperation den Platz tauschen musste.
Fragen-Zähler: Zähler, der aufzeichnet, wie oft Personen nach den jeweiligen Geburtsdaten fragen mussten.
Tiefenzähler: Zähler, der aufzeichnet, wie oft die Struktur insgesamt durchlaufen werden musste (Rekursionstiefe).

Vorgehensweise (Durchspielen des Algorithmus)
Aufbau
1. Experimentalgruppe stellt sich wie im Turnunterricht in einer nach Größe geordneten Riege auf.

Mergesort (Riege)
2. Wenn die Riege aus nur einer Person besteht, ist die Riege sortiert. Also, das Aufteilungsverfahren 2a. FERTIG. Setze bei Schritt 3 fort.

> *Auf das theoretisch und am ersten Blick abstrus wirkende Faktum, dass eine einelementige Liste oder Riege notwendigerweise auch sortiert ist, sollte hier unbedingt hingewiesen werden, damit die Jugendlichen erkennen, dass es stets eines gewissen Nukleus bedarf, damit eine größere, einem bestimmten Kriterium entsprechende Struktur entstehen kann. Wenn man am Ende des Verfahrens prüft, ob und warum wirklich sortiert wurde, wird auf diesen ersten Schritt, der an dieser Stelle zu behandeln ist zurückgegriffen.*

Sonst:

2b. Die Mitte der Riege wird festgestellt und die Riege teilt sich in zwei gleich große Teil-Riegen. Beide Riegen treten nach Teilung einen Schritt zurück (linke nach links hinten, rechte nach rechts hinten, sodass in der Mitte anfangs viel, im Laufe des fortgesetzten Verfahrens immer weniger, aber doch etwas Platz bleibt).

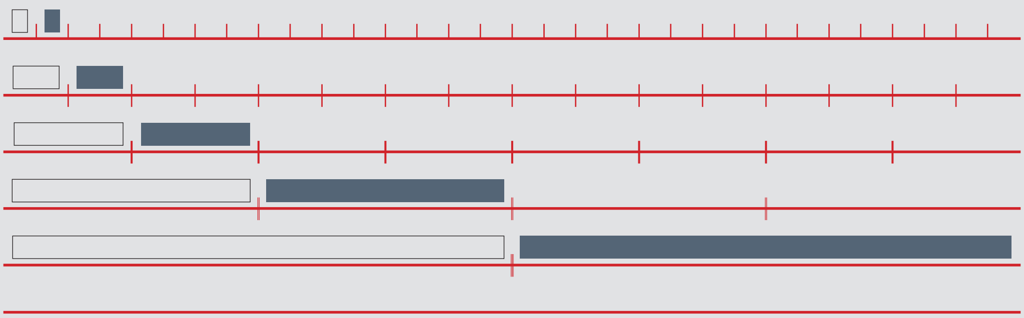
2c. ***Mergesort*** (linke Teilriege)

2d. ***Mergesort*** (rechte Teilriege)

3. ***Mische*** (linke Teilriege, rechte Teilriege) so dass jeweils die Person vom Kopf der jeweiligen Teilriege mit dem „jüngeren" Geburtstag sich an das Ende der vor den beiden Teilriegen stehenden neuen „Gesamt(teil)riege" stellt. Der Kopf der „Gesamt(teil)riege" steht vor der Position auf der der Kopf der linken Teilriege stand oder noch steht.

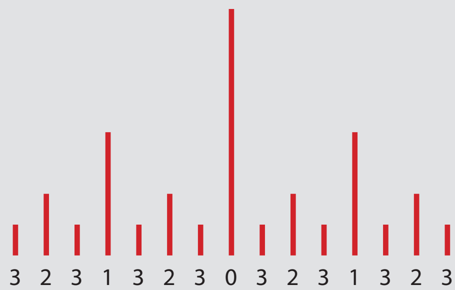
Um den Aufteilungs- und Mischvorgang gut zu beobachten, wird es sinnvoll sein, am Boden Kreidestriche für jene Positionen/Linien zu machen, auf welche die sich teilenden Gruppen zurück treten sollen. Damit hat man auch jene Linie, auf welche die sich mischenden Gruppen vorkommen sollen.

Bodenskizze (Rote Linien mit Andeutung der sich rekursiv teilenden Riege, bis nur mehr Einzelpersonen übrig bleiben.)



In den meisten Klassen wird sich diese zu obiger Beschreibung passende Anordnung empfehlen. Wenn viel Platz vorhanden ist, kann man bei jedem Teilungsschritt die beiden Subriegen auch jeweils rechts und links der zu teilenden (Sub-)riege aufstellen lassen.

Die alternative Anordnung veranschaulicht das Teilungsverfahren etwas besser, insinuiert allerdings, dass zusätzlicher Speicherplatz benötigt wird. Dies ist allerdings nicht der Fall. Beim Verfahren mit Zurücktreten ist es einfacher, nachher feststellen zu lassen, wie viel zusätzlicher Speicherplatz für „Buchhaltungsoperationen" zur Feststellung der Grenzen der eben geteilten Subriegen nötig ist, wenn man nicht zurück tritt (zur Seite tritt) sondern nur die Grenzen der jeweiligen Bereiche markiert werden.



Details zu Mische(linkeTR, rechteT)

A. Die beiden aus dem Aufteilungsteil von Mergesort entstandenen Teilriegen stehen hinter der Linie, auf die sie durch Trennen und Zurücktreten gekommen sind. Die Linie auf der sie vorher standen, ist daher nun frei.

B. Der „Vergleicher" erhebt die Geburtsdaten der beiden an den Kopfpositionen der hinteren Linie stehenden Subgruppen und ersucht jenen mit dem „jüngeren" Geburtsdatum an den Kopf der vorderen Linie zu treten.

C. Nun erhebt der „Vergleicher" weiter die Geburtsdaten der nunmehr an den (reduzierten) Kopfpositionen der hinteren Linie stehenden Personen und ersucht die „jüngere" sich an das Ende der sich eine Linie weiter vorn bildenden Gruppe anzuschließen.

D. Sobald eine der beiden hinteren Gruppen erschöpft ist, kann die verbliebene Gruppe wie sie ist an das Ende der vorderen Linie angeschlossen werden.

Analyse des Algorithmus

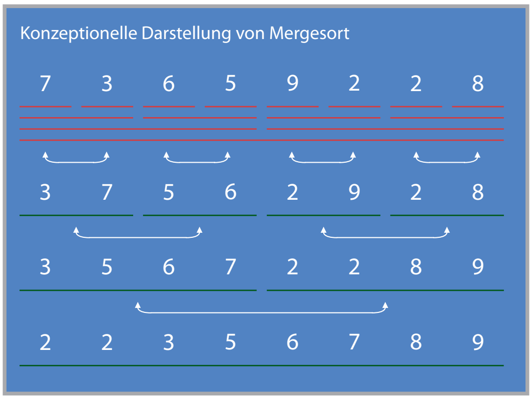
- Welche Beobachtungen wurden von der Beobachtungsgruppe gemacht:

  - Welche Gesetzmäßigkeit gilt zwischen linker und rechter Teilriege?
    A. nach dem Aufspalten?
    B. nach dem Mischen?
  - Welche Gesetzmäßigkeit gilt innerhalb linker und rechter Teilriege?
    A. nach dem Aufspalten?
    B. nach dem Mischen?
  - Endet das Aufspalten in jedem Fall?
  - Endet das Mischen in jedem Fall?
  - Wie viel Linien mussten gebildet werden?
  - Wie viel Fragen mussten beantwortet werden?
    A. pro Gruppe?
    B. über alle Gruppen hinweg in einer Linie?
    C. über alle Linien hinweg?
  - Wie viel Bewegungen mussten durchgeführt werden?
  - Wie lassen sich diese Beobachtungen auf die Zahl der Personen in der Experimentalgruppe umlegen? Wie würden die Zahlen aussehen, wenn in der Experimentalgruppe 10, 100, 1000 Personen gestanden wären?

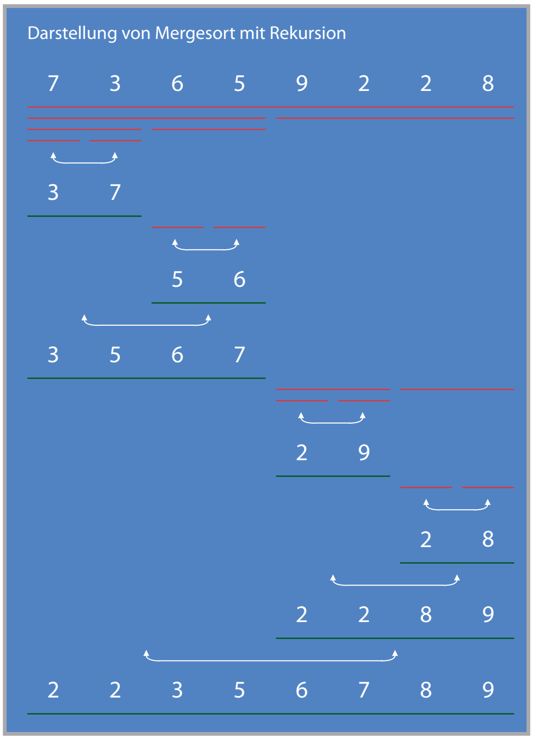
  - Tafelskizze der entstandenen Teilsequenzen

Für die Tafelskizze zu Mergesort bieten sich zwei Varianten an:

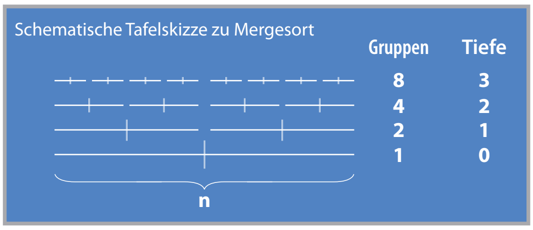
- *Für Klassen unterer Schulstufen, in denen noch nicht programmiert wurde bzw. für Kinder, denen man Rekursion im Detail noch nicht zumuten möchte, bietet sich nebenstehende konzeptionelle Darstellung an.*



> - *Für Oberstufen-Klassen wird es demgegenüber naheliegend sein, das Verfahren (auch) in seiner Verschränkung zwischen teilen und mischen zu zeigen. Diese zweite Form entspricht der angegebenen Lösung in Pseudocode (S. 30). Wichtig ist freilich, dass diese rekursive Lösung schrittweise (hier zeilenweise) als Tafelbild entsteht. Aus der Dynamik der Präsentation können die Jugendlichen die Dynamik des Verfahrens erkennen und auch sehen, dass die reale Lösung letztlich der konzeptuellen Lösung entspricht. Die rekursive Lösung statisch zu projizieren (wie es im Druck leider der Fall ist) verwirrt demgegenüber wohl mehr als es hilft.*



- Anhand einer schematischen Tafelskizze feststellen, dass O(n · ld(n)) Schritte benötigt werden, da die Strukturen wenn man sie gesamtheitlich betrachtet, zwar immer n Elemente haben, aber in die Tiefe gestaffelt, nur ld(n) Ebenen entstehen.



  - o Gilt O(n · ld(n)) immer?
  - o Ist der Aufwand davon abhängig, ob die Riege sortiert war oder ob die Elemente zufällig angeordnet waren?
- Wodurch entsteht dies?

  → „Halbierungen" der Riege, hier immer! (Bei Quicksort nur im günstigsten Fall und auch im Schnitt)

  $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \ldots \frac{1}{2^n}, \ldots$

- Skizzierung des Algorithmus in Pseudocode mit Rekursion und, wenn Zeit ist, auch in Pseudocode mit expliziter Schleifenstruktur. Erkennen der Wiederholungsgesetzmäßigkeiten aus der Rekursion. Erkennen der Endbedingung und der O(ld(n)) Schritte in die Tiefe aus der Rekursion und O(n) Schritte aus der Kombination der Schleifenstruktur.

Pseudocode für Mergesort

Die Idee ist hier, dass ÜL (oder TN) quasi zur Protokollierung des durchgespielten Algorithmus sich den Pseudocode von der Klasse diktieren lässt. Dazu muss man nicht im Vorfeld bereits Programmierung in irgend einer Programmiersprache oder Pseudocode kennen gelernt haben. Die TN fassen dies als eine Art formalisiertes Stichwortprotokoll auf.
Der Algorithmus für Mergesort sollte jedenfalls skizziert werden, jener für Mische ist etwas komplizierter, obzwar das Konzept des systematischen Mischens unmittelbar klar ist. Daher ist er nur dann anzugeben, wenn die Zeit dafür reicht.

```
Mergesort (Feld, links, rechts) {
        WENN links < rechts DANN {
                bestimme mitte;
                Mergesort (Feld, links, mitte-1);
                Mergesort (Feld, mitte, rechts);
        } SONST {} %nur mehr 1 oder kein Element, daher sortiert
        Mische (Feld, links, mitte, rechts, Hilfsfeld);
        überschreibe Feld mit Hilfsfeld
}.
```

```
Mische (Feld, links, mitte, rechts; Hilfsfeld) {
        i := links;
        kl := links; km := mitte;
        SOLANGE kl < mitte UND km ≤ rechts {
                WENN Feld[kl] ≤ Feld[km] DANN {
                        Hilfsfeld[i] := Feld[kl];
                        kl := kl+1
                } SONST {
                        Hilfsfeld[i] := Feld[km];
                        km := km+1
                }
                i := i+1;
        }
        SOLANGE kl < mitte {
                Hilfsfeld[i] := Feld[kl]; kl := kl+1; i := i+1
        }
        SOLANGE km < rechts {
                Hilfsfeld[i] := Feld[km]; km := km+1; i := i+1
        }
}.
```

Vorsicht: Bei rekursiven Algorithmen grundsätzlich gleich Platz lassen für den bzw. die nichtrekursiven Trivialteile. Diese sodann mit Farbkreide einfügen. Man sollte sie auch dann einfügen, wenn sie, so wie oben, leer sind, damit die Jugendlichen diesen Teil nicht vergessen.

- Hinweis auf Rekursivität: Maler, der ein **Selbstportrait** vor seiner Staffelei malt (kann, Moderator einbeziehend, in bis zu 3 Rekursionsschritten auf der Tafel gezeichnet werden). *Eine ausführlichere Behandlung von Rekursion finden Sie in Modul A-Rekursion.*
- Diskussion von Rekursion in Mathematik: $f(n) = n!$

$$f(n) = \begin{cases} 1 & \text{wenn } n = 0 \\ n \cdot f(n-1) & \text{sonst} \end{cases}$$

Weiterführende Literatur

Ottmann Th., Widmayer P.: *Algorithmen und Datenstrukturen*, BI Wissenschaftsverlag, Mannheim, 1993.

Cormen Th. H., Leiserson Ch. E., Rivest R.L.: *Introduction to Algorithms*, MIT Press / McGraw-Hill Book Company, 1989.