## Modul So2 Selection-Sort


**Zeitrahmen**

20 - 30 Minuten (je nach Tiefe der Diskussion)

**Zielgruppe**

- Volksschule, Sekundarstufe I
- Sekundarstufe II als kontrastierendes Sortierverfahren oder in Erweiterung mit Diskussion über Invarianten.

**Lehrziel**

- Vermittlung eines einfachen relativ leicht durchschaubaren Sortierverfahrens.
- Basis für eine Diskussion über die algorithmische Komplexität dieses Verfahrens und Frage, ob es nicht bessere Möglichkeiten geben könnte.

**Requisiten**

Geburtstagsordnung: keine

Sonst: Selbst angefertigte Kärtchen, die einer Halbordnung entsprechen und vor dem Austeilen an die Experimentalgruppe durchmischt werden (zufallsverteilt).

**Partizipanden**

| | |
|---|---|
| **Experimentalgruppe:** | n ≥ 10 Jugendliche |
| **Algoritm. Unterstützung:** | 1 Taktgeber (sichert, dass sich der Algorithmus in beobachtbarer Langsamkeit entwickelt). |
| | 1 Sucher, der jene Person auswählt, die in der noch unsortierten (Rest-) Menge den jeweils kleinsten Geburtstag hat. |
| **Beobachtungs(gruppe):** | |
| Bewegungszähler | Zähler der aufzeichnet, wie oft Personen innerhalb des jeweiligen Durchlaufs den Platz tauschen mussten. |
| Fragen-Zähler | Zähler, der aufzeichnet, wie oft Sucher in den einzelnen Durchläufen nach den jeweiligen Geburtsdaten fragen musste. |
| Tiefenzähler | Zähler, der aufzeichnet, wie oft der Sucher die Struktur insgesamt durchlaufen musste *(Anzahl der Durchläufe der äußeren Schleife)*. |

**Vorgehensweise (Durchspielen des Algorithmus)**

**Aufbau**

1. Experimentalgruppe stellt sich wie im Turnunterricht in einer nach Größe geordneten Riege auf.

**Selection-Sort (Riege)**

2. Wenn die Riege aus nur einer Person besteht, ist die Riege sortiert. Also, das Aufteilungsverfahren ist FERTIG.

*Auf die Sortiertheit der einelementigen Menge muss hier noch nicht besonders abgehoben werden. Wichtig ist hier vielmehr, darauf hinzuweisen, dass jedes Verfahren ein Problem bestimmter Größenordnung benötigt, um überhaupt angewandt zu werden. Beim Sortieren müssen wir vergleichen und dazu bedarf es eben mindestens zweier Elemente.*
*Während diese Überlegung bei rekursiven Verfahren und Bubble-Sort wegen der Endbedingung motiviert ist, ist dies hier eingangs noch nicht der Fall. Daher kann diese für rekursive Verfahren und auch für Bubble-Sort wichtige Feststellung bei Selection-Sort (wie auch bei Insertion-Sort) ggf. an dieser Stelle entfallen, da man in der Experimentalgruppe ja jedenfalls mehr als eine Person hat.*
*Allerdings geht die Abbruchbedingung der äußeren Schleife (Schritt 4) sowie die Diskussion über Invarianz und Variante des Verfahrens letztlich doch wieder davon aus, dass eine ein-elementige Menge (Minimum am Anfang, Maximum am Ende des Verfahrens) sortiert ist. Erweitert man das Verfahren allerdings bis zum Programmieren, ist sie auch wichtig, weil sonst der Aufruf der Sortierprozedur mit einer leeren oder einelementigen Menge unweigerlich zu Problemen führen würde.*

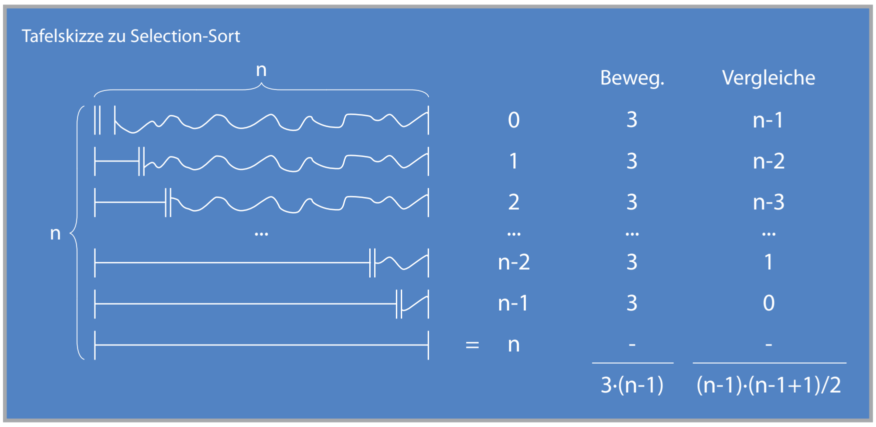
3. Mit der unsortierten (Teil-)Riege werden folgende Anweisungen ausgeführt:
   3.1. Die an der linksten Position stehende Person tritt einen Schritt vor. Damit wird diese Position frei.
   3.2. Der Sucher erfragt das Geburtsdatum der vorgetretenen Person und merkt sich dieses als vorläufig aktuell kleinstes sowie die zugehörige Person als aktuellen Tauschkandidat.
   3.3 Der Sucher wandert sodann am noch unsortierten Teil der Riege nach rechts indem er jede einzelne Person nach dem Geburtsdatum fragt.
      Wenn dieses kleiner ist als das aktuell kleinste, so merkt sich der Sucher nun dieses als aktuell kleinstes sowie die zugehörige Person als aktuellen Tauschkandidat.
   3.4. Sobald der Sucher das Geburtsdatum der am weitesten rechts stehenden Person erfragt hat und auch geprüft hat, ob diese Person allenfalls neuer aktueller Tauschkandidat ist, bricht er seine Wanderschaft ab und ersucht den nunmehr feststehenden aktuellen Tauschkandidaten an das linke Ende der unsortierten (Teil-)Riege zu treten.
   Merke: Es kann auch sein, dass die von der am weitesten links liegenden Position hervorgetretene Person bis zum kompletten Durchlauf der Riege aktueller Tauschkandidat bleibt. Dies ändert nichts am Verfahren, dann tritt eben diese Person in die Lücke ihrer ehemaligen Position zurück.

   *Sollte sich dieser Fall nicht zufällig im Laufe des Gesamtalgorithmus ergeben, empfiehlt es sich, diese Situation mit der Klasse im Rahmen der Diskussion des Verfahrens zu besprechen.*

   3.5. Dadurch ist nun die Position des aktuellen Tauschkandidaten (also der Person mit dem nächst kleinsten Geburtstag) frei geworden. Sie wird von der Person eingenommen, die in Schritt 3.1 die linkste Position frei gemacht hat.

*Bei der Überleitung zum Pseudocode wird zu besprechen sein, dass Schritt 3.4, der hier entfallen kann, wenn die linkste Person auch jene mit dem kleinsten noch verfügbaren Geburtstagsdatum war, durchaus als Dreieckstausch ohne Wirkung durchgeführt wird. Dies ist effizienter als eine explizite Behandlung des Sonderfalls.*
*Die Sonderfallbehandlung entsteht hier lediglich dadurch, dass der Dreieckstausch zwischen Linkstem und Kleinstem auf mehrere Schritte (3.1, 3.4 und 3.5) aufgeteilt wurde. Wir empfehlen dies allerdings, da durch 3.1 die besondere Rolle des jeweils linksten Elements, sowie die Trennung zwischen sortiertem und unsortiertem Teil der Riege für alle unmittelbar ersichtlich wird.*

4. Der Sucher geht nun zur Position jener Person, die unmittelbar rechts von der in 3.4 eingefügten Person ist.
   Wenn diese Person bereits die am rechtesten Rand stehende Person ist, gilt für den unsortierten Bereich die in Schritt 2 zu prüfende Bedingung und wir sind FERTIG;
   Sonst, führen wir das Verfahren ab Schritt 3.1 nochmals durch.
5. Wir beobachten, dass durch die Kombination 3.3 und 3.4 beim ersten Durchlauf dieses Verfahrens die Person mit dem insgesamt kleinsten Geburtstag an den Anfang der Riege gestellt wurde. Ab diesem Zeitpunkt war die Riege stets in einen sortierten und einen noch unsortierten Teil gegliedert, wobei durch jeden Durchlauf der sortierte Teil um eine Person gewachsen, der unsortierte entsprechend um eine Person geschrumpft ist.




**Analyse des Algorithmus**

- Schritt 5 des Verfahrens ist ja kein Verfahrensschritt mehr, sondern ein Analyseschritt. Da davon auszugehen ist, dass nicht alle TN diese Beobachtung gemacht haben, empfiehlt es sich, die Analyse des Algorithmus mit einer Diskussion über diese Beobachtung zu beginnen.

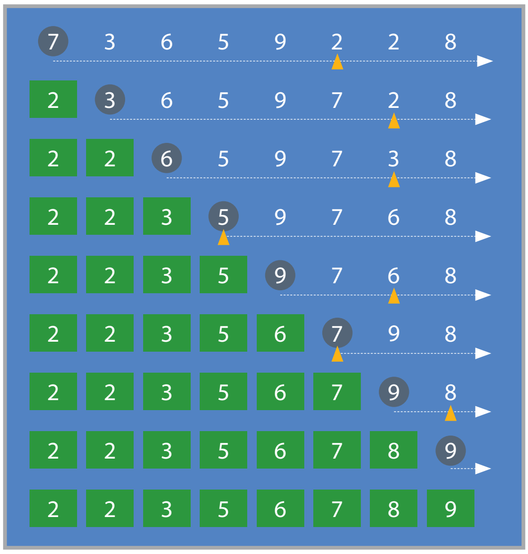
Stimmt diese Aussage/Beobachtung? Warum stimmt sie?
Warum musste im letzten Durchlauf die am weitesten rechts stehende Person nicht mehr geprüft werden? Warum konnten wir ohne weitere Prüfung annehmen, sie sei jene mit maximalem Geburtstagswert?

- Welche Beobachtungen wurden von der Beobachtungsgruppe gemacht:

- Wie viel Fragen mussten beantwortet werden?
- Wie viel Bewegungen haben stattgefunden?
- Hat sich die Anzahl der Bewegungen oder der Fragen im Zuge der Durchläufe geändert? Tafelskizze zur Veranschaulichung!
- Wie oft musste der in 3. beschriebene Ablauf durchlaufen werden? Wird das immer so sein? Wie oft muss er mindestens, wie oft maximal wiederholt werden? Tafelskizze zur Veranschaulichung!
- 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.

*Aus der Tafelskizze (gerade Linien für sortierten Bereich, Wellenlinie für unsortierten Bereich mit schräg absteigender Verkürzung der (unsortierten) Wellenlinien und gleichzeitig wachsenden geraden (sortierten) Linien sollte eine Basis für das Eintragen der Werte der Beobachter und letztlich die Basis für die Erklärung von $O(n^2)$ Abfragen bei $O(n)$ Bewegungen gegeben werden.*

*Wenn sich die Fragen anhand der physischen Übung und der schematischen Tafelskizze nicht ausreichend beantworten lassen, kann man noch eine weiteres Tafelbild mit Zahlen (oder Namen) wie rechts dargestellt nachreichen.*




- Anhand der Tafelskizze feststellen, dass unabhängig vom Grad der Vorsortierung zwischen $O(n^2)$ Abfragen und $O(n)$ Platztauschoperationen benötigt werden.
- Skizzierung des Algorithmus in Pseudocode mit expliziter Schleifenstruktur. Erkennen der Wiederholungsgesetzmäßigkeiten aus den geschachtelten Schleifen. Daraus:
  - o Erkennen der zunehmenden Sortiertheit der Riege und der Rolle der Endbedingung.
  - o Besprechen, warum es sinnvoll ist, den finalen Dreieckstausch auf alle Fälle durchzuführen.
  - o Wenn entsprechende Vorkenntnisse aus Mengenlehre gegeben sind, könnte man anhand der Skizze des Algorithmus in Pseudocode auch in eine Diskussion über Invarianten eintreten.

**Pseudocode für Selection-Sort**

```
Selectionsort (Feld, links, rechts) {
        VAR:    sucher, posAktuellesMinimum: INDEXTYP
                hilfsStelle: FELDELEMENTTYP

        SOLANGE links < rechts {
                posAktuellesMinimum := links;
                sucher := links;
                SOLANGE sucher ≤ rechts {
                        WENN Feld[posAktuellesMinimum] > Feld[sucher] DANN
                                posAktuellesMinimum := sucher;
                        sucher := sucher + 1
                }
                hilfsStelle := Feld[links];
                Feld[links] := Feld[posAktuellesMinimum];
                Feld[posAktuellesMinimum] := hilfsStelle;
        }
}.
```

**Weiterführende Literatur**

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

Li Liwu.: *Java: Data Structures and Programming*; Springer Verlag, 1998.