

# Über Algorithmen

## Aufgabe 4

### Nummer 1

Die systematische Vorgehensweise die Zahlen bzw. uns zu sortieren, also nach dem Prinzip des Bubblesort, ist ein Algorithmus, da in der Aufgabenstellung klar jeder einzelner Schritt beschrieben ist. Außerdem sind diese Schritte auch machbar, da jede Person sich nur bewegen muss um den Platz zu tauschen. Des Weiteren ist die Aufgabe auch nicht unendlich, da man spätestens bei ein Kontrolldurchlauf, bei dem keine Zahlen bzw. Personen mehr getauscht werden, die Aufgabe erledigt/beendet ist. Ein weiterer Grund dafür ist, dass es egal ist welche Personen man dort stehen hatte und auch egal welche Zahlen. Dies hat man daran gesehen, dass es in beiden Gruppen funktioniert hat, die unabhängig voneinander das Problem gelöst haben. Dazu kommt auch, dass zu jedem Zeitpunkt klar war was zu tun ist. Jeder wusste was der nächste Schritt ist. Es gab keine unerklärliche Lücke in der Arbeitsanweisung. Abschließend kann man sagen, dass das Prinzip Von Bubblesort definitiv ein Algorithmus ist, da es alle Merkmale erfüllt.

### Nummer 2

Es ist kein Problem, wenn es mehrere gleiche Zahlen gibt, da diese einfach hintereinander stehen würden:

Beispiel: 7497

1.1 **7**497 (tauschen)

1.2 **4**797 (nicht tauschen)

1.3 4**7**97 (tauschen)

1.4 4**77**9 (man kann nichts tauschen, letzte Zahl)

2.1 **4**779 (nicht tauschen)

2.2 **477**9 (nicht tauschen)

2.3 **4779** (nicht tauschen)

2.4 **4779** (man kann nichts tauschen, letzte Zahl)

→ 7 und 7 stehen nebeneinander

### Nummer 3

Im einfachsten Fall bräuchte man nur einen Durchlauf, bei dem kontrolliert wird ob noch Zahlen getauscht werden müssen. Bei schon sortierten Zahlen wäre das dann nicht der Fall. So wäre ein Durchgang ausreichend.

Beispiel: 5678

1.1 **5678**

1.2 **5678**

1.3 **5678**

1.4 **5678**

#### **Nummer 4**

Damit der Bubblesort-Algorithmus nie terminiert könnte man immer wieder eine Zahl dazu packen. Somit würde es immer weiter gehen.

## **Aufgabe 5**

#### **Nummer 1**

1.1 **371825946**

1.2 **371825946** (tauschen)

1.3 **317825946**

1.4 **317825946** (tauschen)

1.5 **317285946** (tauschen)

1.6 **317258946**

1.7 **317258946** (tauschen)

1.8 **317258496** (tauschen)

1.9 **317258496**

2.1 **317258469** (tauschen)

2.2 **137258469**

2.3 **137258469** (tauschen)

2.4 **132758469** (tauschen)

2.5 **132578469**

2.6 **132578469** (tauschen)

2.7 **132574869** (tauschen)

2.8 **132574689**

2.9 **132574689**

3.1 **1**32574689

3.2 **13**2574689 (tauschen)

3.3 12**3**574689

3.4 123**5**74689

3.5 1235**7**4689 (tauschen)

3.6 12354**7**689 (tauschen)

3.7 123546**7**89

3.8 1235467**8**9

3.9 12354678**9**

4.1 **1**23546789

4.2 **12**3546789

4.3 12**3**546789

4.4 123**5**46789 (tauschen)

4.5 1234**5**6789

4.6 12345**6**789

4.7 123456**7**89

4.8 1234567**8**9

4.9 12345678**9**

5.1 **1**23456789

5.2 **12**3456789

5.3 12**3**456789

5.4 123**4**56789

5.5 1234**5**6789

5.6 12345**6**789

5.7 123456**7**89

5.8 1234567**8**9

5.9 12345678**9**



## Hardcore: Quicksort

Also ich persönlich empfinde Quicksort vom verstehen etwas schwieriger, weil man immer erst nachdenken muss was der nächste Schritt ist. Meinem Gefühl nach zu urteilen, werden bei Quicksort weniger Durchgänge benötigt als bei Bubblesort. in einem Durchgang können nämlich schon mehrere Vergleiche gemacht werden.

## Aufbau Internet

### Aufgabe 1

Meine erste Vorstellung vom weg der Nachrichten



wirklicher Weg von Nachrichten



## Von Informationen, Daten und Codierungen

### Aufgabe 1

In der Gruppe haben wir uns dafür entschieden die Farben anhand eines Tippens auf unserem Rücken zu unterscheiden. Anfangs haben wir unterschiedliche Stärken vom Antippen genutzt, jedoch hat sich später herausgestellt, dass dies in kurzer Zeit schwer war den Unterschied zu spüren. Somit haben wir unsere Strategie geändert und einfach verschiedene Positionen auf unserem Rücken für die verschieden Farben genutzt. Dies klappte besser und war deutlich zeitsparender.

### Aufgabe 2

Bild von unserer Strategie (Partnerarbeit mit Anna):



## Verschlüsselung



## Aufgabe 6

1.

\* der sichere Austausch von dem Schlüssel \* jeder Chat braucht einen neuen Schlüssel damit nicht verschiedene Personen die Chats lesen können.

2.

<b>öffentlicher Schlüssel</b>	<b>privater Schlüssel</b>
wird irgendwo hinterlegt: beispielsweise im Internet	kein Zugriff von anderen
jeder hat darauf Zugriff	für sich behalten, niemanden geben
verschlüsselt Nachrichten, die nur von einem privatem Schlüssel wieder entschlüsselt werden können	entschlüsselt Nachrichten, die mit einem öffentlichen Schlüssel verschlüsselt wurden

3.

\* man verschlüsselt eine Nachricht mit dem öffentlichen Schlüssel wenn man einer Person eine Nachricht schicken möchte \* der private Schlüssel ist zum entschlüsseln der Nachricht damit der Empfänger sie lesen kann

4.

\* man muss nur einmal ein Schlüsselpaar generieren und den öffentlichen Schlüssel irgendwo im Internet hinterlegen. So hat dann jeder Zugriff darauf \* man kann den privaten Schlüssel für jede Kommunikation benutzen und braucht nicht für jeden Chat einen neuen Schlüssel

## Daten in einem Wiki organisieren

Leihvertrag über die Leihe eines mobilen Endgeräts für Schüler\*innen

Zwischen

dem Landkreis Cloppenburg Eschstr. 29 49661 Cloppenburg

vertreten durch den Landrat Johann Wimberg

- im Folgenden Verleiher -

und

—

Name und Anschrift des Schülers/ der Schülerin

vertreten durch:

Name der/ des gesetzlichen Vertreter/ Vertreterin/ Vertreters

# Grafisch programmieren

## Aufgabe 4

Das Blatt Papier erfüllt die Bedingungen eines Algorithmus, da die Farbcodes dem Ozobot eine klare Anweisung geben was er zu tun hat. Die Farben sagen ihm was er zu tun hat und wenn gerade mal keine Reihenfolge an Farben vorgegeben ist sondern nur eine schwarze Linie muss er der nur folgen. Die Reihenfolge von den Farben und deren Bedeutung wurden auf den Ozobot programmiert. Außerdem erfüllen die Farbcodes alle Merkmale eines Algorithmus. Es gibt nicht unendlich viele Farbcodes. Es hat auch jeder Ozobot die gleiche Voraussetzung, da sie alle gleich hergestellt wurden.

# Blockbasiert programmieren

eigene freie Erfindung: zum Ausprobieren



## Aufgaben einfach

### Nummer 1

Beispiel aus dem Video



### Nummer 2

Quadrat



Rechteck



Dreieck



[Code zum alten Bild](#)

codes.ozocode

## Nummer 3

im blauen Kreis bleiben



# Aufgabe mittel

## Nummer 1



From:

<https://cs-free.riecken.de/> - **Informatik 10**

Permanent link:

<https://cs-free.riecken.de/doku.php?id=users:johanna.niemann:start&rev=1704489549>

Last update: **2024/01/05 22:19**

