Über Algorithmen

Aufgabe 4

Nummer 1

Die systematischen 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 Weitern 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, das zu jedem Zeitpunkt klar war was zu tun ist. Jeder wusste was der nächste Schritt ist. Es gab keine unerklärbare 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**7**97 (nicht tauschen)

1.3 47**9**7 (tauschen)

1.4 477**9** (man kann nichts tauschen, letzte Zahl)

2.1 **4**779 (nicht tauschen)

2.2 4**7**79 (nicht tauschen)

2.3 47**7**9 (nicht tauschen)

2.4 477**9** (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 **5**678

1.2 5**6**78

1.3 56**7**8

1.4 567**8**

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 **3**71825946

1.2 3**7**1825946 (tauschen)

1.3 31**7**825946

1.4 317**8**25946 (tauschen)

1.5 3172**8**5946 (tauschen)

1.6 31725**8**946

1.7 317258**9**46 (tauschen)

1.8 3172584**9**6 (tauschen)

1.9 31725849**6**

2.1 **3**17258469 (tauschen)

2.2 1**3**7258469

2.3 13**7**258469 (tauschen)

2.4 132**7**58469 (tauschen)

2.5 1325**7**8469

2.6 13257**8**469 (tauschen)

2.7 132574**8**69 (tauschen)

2.8 1325746**8**9

2.9 13257468**9**

3.1 **1**32574689

3.2 1**3**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 1**2**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 1**2**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.

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 verschiede 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 1

... - .. .-.. .-.. .-. ..- .... - -.. . .-. ... . .

S T I L L R U H T D E R S E E

Unser Übertragungsweg war min einem Kugelschreiber oder generell ein Stift auf den Tisch klopfen. Für den Anfang eines neuen Buchstaben haben wir jeweils 2 Sekunden Zeit dazwischen gelassen. Nach einem Wort haben wir 5 Sekunden Zeit gelassen. Der Empfänger schreibt sich erst den Morsecode, also den Rhythmus auf, bevor er sich die Vorlage zur Hand nimmt und dann die Buchstaben entschlüsselt.

Aufgabe 2

…. .- .-.. .-.. — .– .. . –. . …. - . … -.. .. .-.

H A L L O W I E G E H T E S D I R

Warum es funktioniert: Sowohl Sender als auch Empfänger haben beide die gleiche 'Legende'/ Vorlage und somit können sie genau die Wörter bzw. Sätze miteinander teilen.

Welche Problematik das mit sich bringen kann:

* ungenauer Rhythmus

* zu kurze Pausen

* keine klare Abgrenzung von Wörtern

* Probleme beim Hören des Rhythmus

Aufgabe 3 und 4

1 und 2 Beispiel Material Seite 1

1 bis 3 Aufgaben Material Seite 2

Aufgabe 5

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

Aufgaben einfach

Nummer 1

Nummer 2

Quadrat

Rechteck

Dreieck

Code zum alten Bild

codes.ozocode

Nummer 3

Aufgabe mittel

Nummer 1