

# Mischen possible. Didaktische und unterrichtspraktische Überlegungen zum Mischen im Informatikunterricht.

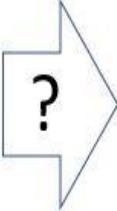
Peter Micheuz<sup>1</sup>

**Abstract:** In diesem Praxisbericht geht es exemplarisch um Zugänge, Werkzeuge und Methoden zum Thema Mischen. Verglichen mit der Vielfalt an Sortieralgorithmen, wird dieses Thema im Bereich digitaler Datenverarbeitung kaum explizit gemacht. Aber warum eigentlich? Dieser Beitrag befasst sich mit der Didaktik des „Anordnens“ von digitalen Objekten als Teil eines zeitgemäßen Informatikunterrichts, in dem unterrichtspraktische Überlegungen und Konzepte mit unterschiedlichen Softwarewerkzeugen und Entwicklungsumgebungen demonstriert und implementiert werden.

**Keywords:** Algorithmen, Computational Thinking, Programmierung, Sortieren und Mischen,

## 1 Einleitung

Es begab sich kürzlich im Informatikunterricht einer 5. Klasse (9. Jahrgangsstufe), in dem das „Schweizer Messer“ unter den Softwarewerkzeugen, nämlich Excel, als Vehikel für informatische Aktivitäten behandelt wurde: Der Autor fragt die Schüler, wie eine nach den Punkten des Biberbewerbs an der Schule absteigend sortierte Liste mit Schülernamen und Klasse wieder schnell „durchmischt“ werden kann, um die Sortierung nach Klasse und Punkten als Übungsaufgabe erneut stellen zu können.



Klasse	Nachname	Vorname	Punkte
1B	Nachname1	Jakob	112
1B	Nachname2	Maja Kathrina	108
1B	Nachname3	Filiz	98
1B	Nachname4	Anna-Giulia	96
1B	Nachname5	Moritz	94
1B	Nachname6	Leonie	91
1B	Nachname7	Fabienne	88
1B	Nachname8	Elias Noel	87
1B	Nachname24	Lara	30
1C	Nachname26	Viktoria	108
1C	Nachname27	Lara	100
1C	Nachname28	Benjamin	92
1C	Nachname29	Gerald Martin	75
1C	Nachname30	Georg Erich	72
1C	Nachname31	Tobias Felix	68

Klasse	Nachname	Vorname	Punkte
5B	Nachname326	David	164
6B	Nachname364	Elias	94
4D	Nachname278	Manuel	148
2A	Nachname113	Kilian Noel	36
2B	Nachname130	Christoph	56
3D	Nachname219	Maximilian	100
5A	Nachname314	Minea	100
1C	Nachname26	Viktoria	108
2C	Nachname151	Melissa Chiara	70
2D	Nachname175	Daniel Luca	63
3D	Nachname212	Mikka	128
1B	Nachname24	Lara	30
2A	Nachname107	Shu Yi	84
2A	Nachname103	Kimmy Gloria Maria	92
1D	Nachname54	Lin	84
4F	Nachname262	Lukas	41

Abbildung 1: Von der Ordnung zur Unordnung

Ich habe nicht schlecht gestaunt, als mir gleich mehrere Schüler, ohne lange überlegt zu haben, folgende Lösung vorschlugen: „Eine leere angrenzende Spalte mit Zufallszahlen

<sup>1</sup> Alpen-Adria Gymnasium Völkermarkt, Pestalozzistraße 1, A-9100 Völkermarkt, peter.micheuz@aon.at

befüllen, die Daten nach diesen Zufallszahlen sortieren und anschließend die Spalte mit den Zufallszahlen wieder löschen“. Auf meine Frage, wie sie so schnell draufgekommen sind, gab es eine überraschende Antwort: „Sie haben es uns vor einiger Zeit einmal so nebenbei gezeigt ...“.

Auf meine weitere Frage, warum die Entwickler von Microsoft nicht an einen „Mischbutton“ neben den Sortieroptionen in den komplexen Menübändern gedacht haben, gab es



noch keine schlüssigen Antworten. Diese gab es (auch noch) nicht auf die Fragen, wie im Computer Pseudozufallszahlen erzeugt werden, und wie die Inhalte der Zellen so angeordnet bzw. zielgerichtet vertauscht werden, dass die Werte in den Zellen auf- oder absteigend sortiert sind.

Im Fokus der weiteren Überlegungen liegt in diesem Beitrag, bis auf die Erwähnung des Mergesort, nicht das zielgerichtete Sortieren, sondern die zufallsgesteuerte Produktion von Unordnung.

## 2 Lottozahlen

Wenn von Zufall und unsortierten Zahlen die Rede ist, kommen einem fast unweigerlich Lottozahlen in den Sinn. Die offene Aufgabe, wie einem Computer beizubringen ist, Lottozahlen zu erzeugen, wurde im Informatikunterricht mit Sicherheit schon tausende Male gestellt und mit verschiedenen Werkzeugen, Entwicklungsumgebungen, Bibliotheken, unterschiedlichen Algorithmen und Datenstrukturen, und mehreren Programmierparadigmen sowie auf diversen Abstraktionsebenen behandelt und realisiert. Eine auch nur annähernd erschöpfende Behandlung all dieser Zugänge ist an dieser Stelle nicht möglich, aber ein Anknüpfungspunkt und eine Analogie zum Verfahren in der Einleitung soll näher beschrieben werden. Warum nicht die Lottozahlen nach der „Katalysatormethode“ (temporär angelegtes Feld mit Zufallszahlen, siehe Einleitung) als Mischung von sechs unterschiedlichen Zahlen in einem bestimmten Zahlenbereich (in Deutschland 1-49, in Österreich 1-45) erzeugen?

Abb. 2 zeigt die spaltenmäßige Anordnung (Spalte A: Lottozahlen, Spalte B: Zufallszahlen) des „einfachen“ Lottozahlengenerators, der mittels der Neuberechnungstaste F9, dem Selektieren der beiden Spalten und der richtigen Anwendung des Sortierassistenten die Lottozahlen in den Zellen B2:B7 anzeigt. Auf die Möglichkeit der Automatisierung dieser wenigen linearen Handlungsschritte in Form eines „Excel-Makros“ soll an dieser Stelle nicht eingegangen werden, obschon „Makroprogrammierung“ eine empfehlenswerte Möglichkeit darstellt, einen Grundpfeiler der Informatik, nämlich die Automatisierung, schrittweise erfahrbar zu machen.

Sortieren nach Spalte C

	A	B	C	D	E	F	G
1							
2		1	0,44063		2		26
3		2	0,02593		31		42
4		3	0,74237		44		12
5		4	0,34609		37		30
6		5	0,17009		34		34
7		6	0,61382		21		18
8		7	0,70308				
9		8	0,87186				
10		....	....				
11		....	....				
12		44	0,96311				
13		45	0,54996				

=RANG(C2;\$C\$2:\$C\$46)

VERGLEICH(KKLEINSTE(C\$2:C\$46;ZEILE());  
\$C\$2:\$C\$46;0)

Abbildung 2: Varianten von Lottozahlengeneratoren in MS-Excel

Einen vertieften und lohnenden Ausblick auf den „Zauber“ [FL15] funktionaler Modellierung und deklarativer Programmierung, die allein mit der zielgerichteten und kompetenten Anwendung der Bibliotheksfunktionen von MS-Excel (oder ähnlicher Tabellenkalkulationssoftware) möglich sind, bieten die beiden anderen Varianten der Lottozahlenerzeugung (siehe Abb. 2 rechts) mittels der verketteten Funktionen VERGLEICH, KKLEINSTE, ZEILE sowie der Verwendung der algorithmisch interessantent Funktion RANG.

### 3 Mischen in der Praxis – „Real World“-Anwendungen

Bevor wir uns mit den für das Mischen notwendigen Elementaralgorithmen und Daten(strukturen) beschäftigen, sei hier eine weitere - möglicherweise unter Informatiklehrenden wenig bekannte und daher möglicherweise exotische - (My)SQL-Anweisung zur Produktion von sechs Lottozahlen angeführt:

```
select zahlen limit 6
from lottozahlen
order by rand()
```

Die Beschreibung dieses Mischbefehls in einem MySQL-Tutorial im Web [MY19] in Abb. 3 führt uns wieder zum Konzept und die Methode der Sortierung nach Zufallswerten zurück und demystifiziert diesen scheinbar widersprüchlichen Befehl „order by rand()“. Diese Widersprüchlichkeit gibt es vordergründig auch beim Mergesort („Sortieren durch Verschmelzen/Mischen“).

### MySQL Select Random Records - MySQL Tutorial

[www.mysqltutorial.org](http://www.mysqltutorial.org) > [MySQL Tips](#) ▾ [Diese Seite übersetzen](#)

MySQL select random records using ORDER BY RAND() The RAND() function generates a random value for each row in the table. The ORDER BY clause sorts all rows in the table by the random number generated by the RAND() function. The LIMIT clause picks the first row in the result set sorted randomly.

Abbildung 3: Beschreibung der ORDER BY RAND()- Anweisung im Web

An dieser Stelle darf nicht verschwiegen werden, dass dieser SQL-Befehl ein bestimmtes Maß an wohlvorbereitem „Umfeldwissen“ in Form von konkreten Werkzeugkenntnissen voraussetzt, wenn er sinnvoll z.B. in einem zeitgemäßen Web-User-Interface eingesetzt werden soll. Das V (Verarbeitung) im EVA-Prinzip ist mit dem obigen SQL-Befehl der leichtere Teil der Übung, weil er als Blackbox verwendet wird und die Realisierung des zugrundeliegenden Misch-Algorithmus bisher noch kein Thema war. Sowohl die Eingabe der Daten als auch die Ausgabe der Daten setzen viel Unterrichtsarbeit voraus, wenn von den Schülern aktive Entwicklerleistungen in Form von Modellierung, Programmierung und dem Gestalten von bereits einfachen Benutzerschnittstellen eingefordert werden.

Anhand von drei verschiedenen Zufallsfragen, die aus einem größeren Fragepool auf einem Webportal nach jedem Seitenaufruf neu generiert werden, soll dies illustriert werden.

- ▶ Was bedeutet True Color? Was bedeutet RGB im Zusammenhang mit Bildschirmfarben?
- ▶ Welche Anwendungsprogramme gibt es? Ist ein Betriebssystem auch ein Anwendungsprogramm?
- ▶ Was bedeutet "etwas zippen"? Wann macht es besonders viel Sinn, wenn etwas gezippt wird?

Für diese Ausgabe ist das folgende „Codeschnipsel“ auf einem Webserver als „Mischung“ von HTML, PHP und (My)SQL verantwortlich:

```
mysql_select_db("unterstufe");
$abfrage=mysql_query("select frage
                    from basisfragen
                    order by rand()");
for ($i=0;$i<3;$i++) {
    $row = mysql_fetch_row($abfrage);
    echo("<center><b><img src='pfeil.gif'>".$row[0]."
    </b></center>");
}
```

Auf der gleichen Portalseite [SI19] werden nicht nur Texte durchmischt, sondern nach jedem Seitenaufruf drei verschiedene Hardwarebausteine angezeigt (siehe Abb. 4). Diese Anzeige wird aus didaktischen Gründen unter Ausnutzung einer der vielen Bibliotheksfunktionen und der Datenstruktur ARRAY der serverseitigen und weitverbreiteten Programmiersprache PHP realisiert.

---

... viele Hardwarebausteine



Abbildung 4: Zufällige Anordnung von Bildern

Im folgenden kurzen, aber gehaltvollen und leistungsfähigen serverseitigen HTML-PHP-Skript ist der SHUFFLE-Befehl als PHP-Bibliotheksfunktion nicht zu übersehen:

```
echo("<p><h2>... viele Hardwarebausteine</h2>");
$zz = range(0,18);
shuffle($zz);
$bild1="bilder/".$zz[0].".gif";
$bild2="bilder/".$zz[1].".gif";
$bild3="bilder/".$zz[2].".gif";
echo("<p><center><img src='\$bild1'>
      <img src='\$bild2'>
      <img src='\$bild3'>
    </center></p>");
```

Puristen und Dogmatiker unter den Informatikern werden einwenden, dass Skriptsprachen wie PHP, mit ihrer schlampigen Art mit Datentypen umzugehen, zu einem schlechten Programmierstil verleiten. Ich lasse das unkommentiert stehen. Der geneigte Leser möge sich dazu selbst ein Urteil bilden, sich an Expertenforen wie z.B. quora.com beteiligen und daraus die Konsequenzen für den (eigenen) Informatikunterricht ableiten.

Die Anweisung `$zz = range(0,18)` legt ein Array `$zz[0]=0, $zz[1]=1, ... $zz[18]=18` an, der Befehl `shuffle($zz)` durchmischt zufällig die Inhalte der Felder, so einfach ist das! Die hier nicht offensichtlichen Tücken des Skripts liegen weniger darin, die Wirkung des Shuffle-Befehls zu verstehen. Probleme bereiten da schon eher die kunstvolle Anwendung („Maskierung“) der Anführungszeichen und das Konzept von externen, auf dem Server im Ordner „bilder“ gespeicherten Hardware-Bildern im GIF-Format (0.gif, 1.gif, ... 18.gif) sowie das Konzept ihrer internen Anbindung an das Array `$zz`.

Nach diesem kurzen Ausflug in die Webprogrammierung inklusive werkzeugabhängiger Zugänge zum Mischen von Zahlen, Text und Bildern mit der „Katalysatormethode“ (siehe oben), der (My)SQL-Anweisung „order by rand()“ und dem Shuffle-Befehl aus der PHP-Funktionsbibliothek, verlassen wir die Komfortzone der beschreibenden Methode und betreten die prozedurale Welt elementarer Algorithmen.

## 4 Mischen als zufälliges Tauschen von Variableninhalten

Dieser Beitrag darf auch als späte und logische Fortsetzung eines INFOS-Beitrages „Ein kleiner Schritt für LehrerInnen, ein großer Schritt für SchülerInnen“ aus dem Jahr 2005 gesehen werden [MI05], in dem es im 6. Kapitel „Ringtausch und Umfüllaufgabe“ hieß: „In diesem Kapitel ist nur angedeutet worden, welch reichhaltiges, didaktisch wertvolles Potential bereits in (einfachen) Aufgabenstellungen - wie etwa dem Tausch der Inhalte zweier Variablen - steckt. Die Legitimation, den "Platztausch" im Informatikunterricht zu thematisieren, ist aber auch durch unzählige lebensweltliche Anwendungen und die Tatsache gegeben, dass er das Fundament vieler Algorithmen darstellt, in denen es um Misch- sowie Sortierverfahren geht.“

Beginnen wir mit einer konkreten Programmieraufgabe, diesmal in Javascript, einer Sprache, die derzeit einen wahrhaftigen Boom erlebt und an der man nicht vorbeikommt, wenn es um die Entwicklung von Frontend-Anwendungen im Web geht. Unsere Aufgabenstellung lautet so: „Eine bestimmte Anzahl an Schülerbildern sind auf einer Schulwebsite in



tabellarischer Form nach jedem Seitenaufruf durchmischt anzuordnen.“

Anmerkung: Die vier Menschen links existieren in Wirklichkeit nicht. Sie sind der auf Artificial Intelligence und Deep Learning-Algorithmen beruhenden Website „thispersondoesnotexist.com“ entnommen.



Im Unterschied zur serverbasierten Skriptsprache PHP verfügt die Programmiersprache Javascript zwar über eine Methode „sort“ der Objektklasse „array“, aber leider (oder aus didaktischer Sicht Gottseidank) keine Methode „shuffle“, die es mit einer Anweisung gestatten

würde, die Bildnummern zu durchmischen. Es wird an dieser Stelle auch von der Methode des „Katalysatorfelds“ (Sortierung eines temporären Zufallsfeldes) Abstand genommen. Der Leser ist eingeladen auszuprobieren, ob dieser Workaround einfach zu realisieren ist.

Stattdessen bewegen wir uns (notgedrungen) auf die Ebene des „Dreiecks- bzw. Ringtausch“ und realisieren den Mischvorgang durch wiederholtes Tauschen der Array-Inhalte. Wenn es in einer Entwicklungsumgebung keine (Bibliotheks)Funktion oder keine entsprechende Methode gibt, steht es jedem frei, sich im algorithmischen Denken und in der Implementierung derselben zu versuchen. Das Kernstück des folgenden HTML-Files, das vollständig in Javascript codiert und lauffähig ist, wenn die Fotos im gleichen Ordner wie das HTML-Skript zur Verfügung stehen, ist die Funktion „shuffle“ mit dem Array „fotos“. Dieses dient sowohl als Eingabeparameter als auch als Ausgabefeld.

```
<script>
function shuffle(fotos) {
    var laenge = fotos.length, temp, index;
    while (laenge > 0) {
        index = Math.floor(Math.random() * laenge);
```

---

```

        laenge = laenge - 1;
        temp = fotos[laenge];
        fotos[laenge] = fotos[index];
        fotos[index] = temp;
    }
    return fotos;
}

var bilder = new Array()
bilder[0] = 'name_a.jpg'
bilder[1] = 'name_b.jpg'
bilder[2] = 'name_c.jpg'
bilder[3] = 'name_d.jpg'
shuffle(bilder);
var p = bilder.length;
for (var i = 0; i <= 3; i++)
    { var bildnummer = i;
      document.write('<img src="'+bilder[bildnummer]+'>');
      if (i % 2 == 1) {
          document.write('<br>');
      }
    }
</script>

```

Der in der Funktion „SHUFFLE“ implementierte Algorithmus ist optimal mit einer Komplexität von  $O(n)$  und wird auch als Fisher-Yates-Shuffle bezeichnet.

```

function randperm(n)
P = [1:n]      // Initialisierung mit der identischen Permutation
for i = n downto 1 // Schleife über die Einträge von P
    z = random(i) // Gleichverteilte Zufallszahl zwischen 1 und i
    swap(P(i),P(z)) // Vertausche die Zahlen an den Stellen i und z
end
return P

```

Abbildung 5: Pseudocode, Quelle: [https://de.wikipedia.org/wiki/Zufällige\\_Permutation](https://de.wikipedia.org/wiki/Zufällige_Permutation)

Ronald Aylmer Fisher (1890-1962) und Frank Yates (1902-1994) waren bedeutende Statistiker des 20. Jahrhunderts. Dass das optimale Mischen nach ihnen benannt ist, ist kein Zufall. Zufallsanordnungen, mathematisch ausgedrückt zufällige Permutationen, werden unter anderem bei der Analyse von Sortierverfahren, in der Kryptographie, bei randomisierten Algorithmen und vor allem bei Simulationen benötigt.

Übrigens eignet sich der Fisher-Yates-Algorithmus auch vorzüglich für die (optimale) Erzeugung von Lottozahlen, wenn diese mit der Datenstruktur Array realisiert wird. Dies passiert weltweit in tausenden Lottoannahmestellen beim Kauf von computergenerierten Zufallstipps.

## 5 Simulationen und weitere Unterrichtsbeispiele

Interessante Anmerkungen zum Bildungswert der Informatik finden sich bei Rechenberg [RE94], der im Kapitel „Informatik und Kultur“ vier wesentliche Bereiche der Informatik herausstreicht. Erstens das algorithmische Denken, das beim Verstehen von Konzepten jeder Art hilft [vgl. KN74], aber „keinem auf Anhieb gelingt, und mühsam mit großer Geduld erworben werden muß“, zweitens das Studium und die Meisterung komplexer Zusammenhänge, und drittens, dass der Computer „ein so vielseitiges Instrument zur Simulation und zum Ausführen von Gedankenexperimenten ist“. Der vierte Bildungswert der Informatik besteht für Rechenberg darin, zu zeigen, was man mit Computern vor allem nicht automatisieren kann. Damit hat er nicht zuletzt die aktuelle Diskussion um die (Auswirkungen der) Artificial Intelligence bereits vor 25 Jahren vorausgesehen.

Im Folgenden werden Aufgaben und Anregungen zum Thema Mischen im Zusammenhang mit einfachen Simulationen und unterrichtspraktischen Anregungen gestellt.

### Sitzanordnungen und BOGO-Sort

Die Antwort auf die Frage, wie viele Sitzanordnungen es in einer Klasse mit z.B. 25 Stühlen gibt, verblüfft nicht nur die SchülerInnen, die sich - das erste Mal mit dieser Frage konfrontiert - erfahrungsgemäß extrem verschätzen.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	For stelle = 25 To 1 Step -1 zufall = Int(stelle * Rnd) + 1 h = Cells(1, zufall) Cells(1, zufall) = Cells(stelle) Cells(stelle) = h Next
D	E	K	A	B	O	J	L	C	T	W	H	U	M	G	N	F	Y	I	X	P	Q	V	S	R	
J	U	C	L	K	V	Q	G	T	W	I	O	R	H	D	E	P	X	N	A	S	M	Y	F	B	
E	R	S	F	Y	N	P	I	L	C	T	V	B	O	A	K	Q	G	J	D	H	U	X	M	W	
O	A	C	R	K	Q	E	N	F	V	U	P	X	G	S	L	D	J	I	Y	H	W	B	M	T	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	

Abbildung 6: Permutation von 25 Buchstaben

Die Zahl  $25! = 15.511.210.043.330.985.984.000.000$  sprengt das menschliche Vorstellungsvermögen und sollte einen ambitionierten Klassenlehrer davor bewahren, in einem Schuljahr alle Mischvarianten physisch ausprobieren zu wollen, und wohl auch den Informatiklehrer, alle Permutationen digital anzuzeigen, geschweige denn ausdrucken zu wollen. Sehr wohl können die Computer im Informatikraum damit beschäftigt werden, alle(!) Buchstaben-Mischungen zu produzieren und darauf zu warten, bis der Buchstabensalat wieder sortiert ist. Dieses Sortierverfahren - solange zufällig mischen, bis alle Objekte sortiert sind - führt laut Wikipedia (<https://de.wikipedia.org/wiki/Bogosort>) die Bezeichnungen Bogosort, Alexsort, Monkeysort oder Stupidsort. Im krassen Unterschied dazu sei hier das hocheffiziente Mergesort (Sortieren nach dem Prinzip Teile und Herrsche mit anschließendem Verschmelzen/Mischen) angeführt, zu dem es mit Stand 10.2.2019 über eine Million Google-Treffer mit unzähligen guten Beschreibungen und Erklärvideos gibt.

### Magisches Quadrat und eine Biber-Aufgabe

Fachdidaktisch und methodisch interessant im Zusammenhang mit dem Mischen ist auch die Suche nach dem „Magischen Quadrat“ in allen Varianten (siehe Abb. 7).



Abbildung 7: Magisches Quadrat und Biber Aufgabe

Die Abbildung links zeigt eine von unzähligen Implementierungen des Magischen Quadrats (hier in Excel). Die Benutzerschnittstelle mit vier Buttons und einigen berechneten Feldern weist auf einen interessanten didaktischen Zugang zur Lösung dieser Aufgabe hin. Es handelt sich hier um eine Mischform der von Excel bereitgestellten Funktionen (verdeckte Algorithmen: Summen, Minimum und Maximum) und in VBA (Visual Basic for Applications) codierten Tausch- und Mischalgorithmen, die im Hintergrund arbeiten.

Biber-Aufgaben bieten vielfältige Anknüpfungspunkte für informatisches Denken und Handeln, so auch exemplarisch die in Abb. 6 dargestellte originelle „Sortieraufgabe“. In der Grundschule sollten die Balken/Stäbe haptisch (aus)sortiert werden, in der Sekundarstufe I bietet sich funktionale Modellierung an, und in der Sekundarstufe II ist eine Simulation inklusive Programmierung und Codierung empfehlenswert.

Was hat das mit dem Mischen zu tun? Sehr viel. Warum nicht computergestützte Experimente automatisiert durchführen, die den Zusammenhang zwischen der Anzahl von Dachhölzern und allen(!)  $11! = 39.916.800$  möglichen Anordnungen herstellen?

## 6 Schlussanmerkungen

Dem aufmerksamen Leser wird sicher aufgefallen sein, dass bisher ein weitverbreiteter lebensweltlicher Aspekt des Mischens ausgespart wurde. Es geht um die Kultur der Kartenspiele, bei denen der Zufall und das Kartenmischen eine erhebliche Rolle spielen. Bei dieser haptischen Erfahrung ist die wohl weitverbreitetste Methode jene des „Packelns“ (entspricht dem mehrmaligen Abheben), bei dem mit beiden Händen öfters jeweils zwei nach Gefühl unterschiedliche Stapel vertauscht werden. Dabei soll es schon vorgekommen sein, dass nicht ordentlich gemischt und getrickst wird. In diesem Zusammenhang sei auf eine ausführliche Quelle für Kartentricks mit informatischem Background „Die Magie der Informatik: Kartentricks“ [CS19] verwiesen, in der u.a. auch der Trick „Perfektes Mischen“ vorgestellt wird.

Kartenspielen und haptisches Kartenmischen sollten im Unterricht der Primarstufe einen Fixplatz haben. Später in der Sekundarstufe I kann die Haptik z.B. in einer Tabellenkalkulation als Teil einer Bedienungsschulung (Verschieben, Kopieren, siehe Abb. 8) simuliert

1	2	3	4	5	6	7	8	9	10
5	6	7	8	9	10	1	2	3	4
8	9	10	1	2	3	4	5	6	7

Abbildung 8: Simulation des Abhebens

und dabei versucht werden, mit möglichst wenig Mischungsvorgängen eine gute Durchmischung zu erreichen. Die programmiertechnische Realisierung und die Diskussion, was „gut durchmischt“ heißt, kann in der Sekundarstufe II weitergeführt werden.

## Literaturverzeichnis

- [CS19] Die Magie der Informatik: Kartentricks im Rahmen des CS4FN-Projekts (Computer Science for Fun), online unter: <http://www.cs4fn.org/magie>, Stand 10.2.2019
- [FL15] Fleckenstein J. u.a.: Excel – Das Zauberbuch. Markt&Technik, 2015.
- [KN74] Knuth D.E: Computer Science and its Relation to Mathematics. American Mathematical Monthly 1974.
- [MI05] Micheuz P.: Ein kleiner Schritt für LehrerInnen, ein großer Schritt für Schüler. In: Steffen F.: Unterrichtskonzepte für informatische Bildung, INFOS-Tagungsband, 11. GI-Fachtagung Informatik und Schule, 28.-30. September 2005.
- [MY19] MySQL-Tutorial: <http://www.mysqltutorial.org/select-random-records-database-table.aspx>, Stand: 10.2.2019
- [RE94] Rechenberg P.: Was ist Informatik? Hanser-Verlag, 2. Auflage, 1994.
- [SI19] Portal Schulinformatik: <https://www.gym1.at/schulinformatik/unterstufe>, Stand 10.2.2019