Unterrichtssequenz zum Thema «Induktion» - Einführung

Diese Unterrichtssequenz soll als Einführung in die vollständige Induktion als Beweismethode betrachtet werden. Diese Methode wird zuerst in einem mathematischen Kontext eingeführt. Nachfolgend wird gezeigt, dass sie als Verfahren in der Informatik angewendet werden kann, und zwar als Werkzeug für die Analyse von Algorithmen. Die Aufgaben sind bewusst eher einfach gehalten, sowohl aus mathematischer als auch aus informatischer Sicht, damit sich die Schüler und Schülerinnen (SuS) auf die technischen Aspekte der Induktion gut konzentrieren können und verstehen, wie nützlich dieses Beweisverfahren in unterschiedlichen Kontexten sein kann.

1. Vorwissen

<u>Vorwissen aus der Mathematik:</u>

Der Umgang mit Potenzen, mit Folgen und mit Reihen ist den SuS bekannt.

<u>Vorwissen aus der Informatik:</u>

Die SuS haben idealerweise schon einige Sortierverfahren kennengelernt. Sie sind vertraut mit dem Konzept der Laufzeit eines Algorithmus und können anhand eines Pseudocodes oder echten Codes (z.B. in Python) nachvollziehen, wie man die Laufzeit eines Algorithmus abschätzen kann, indem man die relevanten Einzeloperationen wie Vergleiche, Vertauschungen, Variablenzuweisungen, … zählt.

2. Zielsetzungen des Unterrichtsequenzen

- Die SuS entdecken die vollständige Induktion als Beweis- und Argumentationsmethode im Kontext reiner mathematischen Probleme, aber auch im Kontext der in Informatik, z.B. als Instrument, um Algorithmen zu untersuchen.
- Die SuS kennen die Struktur eines Beweises per Induktion und können erklären, welche Rolle dabei die Induktionsvariable, die Induktionsverankerung, die Induktionsannahme und der Induktionsschritt spielen.
- Die SuS können nachvollziehen, wann man ein Problem per Induktion lösen kann, und können die Induktionsbeweise auch korrekt durchführen.

3. Unterlagen

Wie könnte man die folgende Aussage überprüfen?

„Die Summe der ersten n ungeraden natürlichen Zahlen ist gleich dem Quadrat der Anzahl Summanden.“ *(Franciscus Maurolicus, 1494-1575)*

Es ist klar, dass man diese Aussage nicht **direkt** für jeden Wert von n überprüfen kann, da man in dem Fall unendlich viele Überprüfungen machen müsste. In dieser Lektion lernst Du eine Beweismethode kennen, **die vollständige Induktion**, die es mithilfe einer geschickten Argumentation ermöglicht, diese Aussage (und noch viele andere) rigoros für unendlich viele Fälle zu überprüfen.

Wir übersetzen zunächst die Aussage in die mathematische Sprache. Man muss für jedes $n \in \mathbb{N}$ beweisen, dass die Summe

$$s_n = 1 + 3 + 5 + \cdots + (2n - 1) = \sum_{i=1}^{n} (2i - 1)$$

gleich n^2 ist. Die Variable n heisst die **Induktions-Variable**. Sie spielt eine zentrale Rolle im Beweisvorgehen.

Konzeptuell gesehen funktioniert die Beweistechnik der vollständigen Induktion etwa wie das Hochsteigen einer (unendlichen) Treppe. Wenn man vom Erdgeschoss in die nächsten Stockwerke will, soll man zuerst die erste Treppenstufe besteigen. Und angenommen, man ist fähig, von einer *beliebigen Treppenstufe* n auf die *nächste* Treppenstufe $n + 1$ zu steigen, schafft man es sicher, die *ganze* Treppe zu besteigen, auch wenn sie unendlich viele Stufen hätte.

Bildquelle: http://clipart-library.com/clipart/144246.htm

Mathematisch gesehen bedeutet das:

1. Man zeigt zuerst, dass die zu beweisende Aussage $A(n)$ für einen *initialen* Wert von n, n_0, wahr ist. Das ist der Schritt der **Induktionsverankerung.** In der Praxis ist oft $n_0 = 0$ oder 1.
2. Danach nimmt man an, dass die Aussage $A(n)$ für einen beliebigen Wert der Induktionsvariablen n wahr ist. Das ist der Schritt der **Induktionsannahme (IA).**
3. Mithilfe der Induktionsannahme versucht man zu zeigen, dass dann die Aussage $A(n + 1)$ für den nächsten Wert $n + 1$ gilt. Das ist der **Induktionsschritt** $A(n) \to A(n + 1)$.

Wenn man die Punkte 1-3 geschafft hat, ist die allgemeine Gültigkeit der Aussage $A(n)$ für alle $n \geq n_0$ bewiesen.

Aufgabe 1

Überlege Dir (eventuell in Partnerarbeit) und erkläre, wieso die Schritte 1, 2, 3 reichen, um die Aussage $\mathrm{A(n)}$ tatsächlich für alle Werte von $\mathrm{n} \in \mathbb{N}$ zu beweisen. Wieso ist man sicher, dass z.B. $\mathrm{A}(\mathrm{n}_0 + 1)$ gilt? Wieso kann man sicher sein, dass es keine «Lücken» gibt, d.h. keine einzelnen gestreuten Werte von n, für die $\mathrm{A(n)}$ nicht gilt?

Lösung zur Aufgabe 1:

*Man überprüft explizit, dass $A(n_0)$ gilt, und dass, **wenn** $A(n)$ gilt, **dann** $A(n+1)$ gilt für eine beliebige natürliche Zahl $n \geq n_0$. Konkret wird deshalb $A(n_0+1)$ sicher gelten, weil die Gültigkeit von $A(n_0)$ die Gültigkeit von $A(n_0+1)$ impliziert. Und weil $A(n_0+1)$ gilt, wird auch sicher $A(n_0+2)$ gelten, usw. Dies gilt für alle Werte von n der Reihe nach, und es kann nie eine Lücke vorkommen (sonst hätte man logische Probleme beim Induktionsschritt, der nur für spezielle Werte von n funktionieren würde).*

Testen wir die Induktionsmethode nun an unserem Problem und beweisen wir, dass für jedes $n \in \mathbb{N}$ gilt:

$$s_n = 1 + 3 + 5 + \cdots + (2n-1) = \sum_{i=1}^{n} (2i-1) = n^2$$

1. **Induktionsverankerung:** Hier $n_0 = 1, s_{n_0} = s_1 = 1$ und für $n = 1$ gilt $n^2 = 1^2 = 1$, also ist die Induktionsverankerung $s_1 = 1^2$ wahr.
2. **Induktionsannahme (IA)**: Wir nehmen an, dass $s_n = \sum_{i=1}^{n}(2i-1) = n^2$ wahr ist.
3. **Induktionsschritt** $s_n \to s_{n+1}$: Untersuchen wir nun, ob die Aussage auch für $n+1$ gilt:

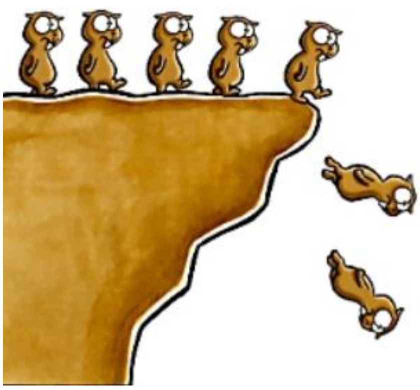
$$s_{n+1} = \sum_{i=1}^{n+1} (2i-1) = \sum_{i=1}^{n} (2i-1) + 2 \cdot (n+1) - 1$$
$$= \boldsymbol{s_n} + 2 \cdot (n+1) - 1 \stackrel{IA}{=} \boldsymbol{n^2} + 2 \cdot (n+1) - 1 = n^2 + 2n + 1 = (n+1)^2$$

4. Somit wurde die Gültigkeit der Aussage

$$1 + 3 + 5 + \cdots + (2n-1) = \sum_{i=1}^{n} (2i-1) = n^2$$

mithilfe der vollständigen Induktion für alle $n \geq 1$ erfolgreich bewiesen.

Aufgabe 2




a. Die Lemming-Comicfiguren sieht man oft reihenweise über Klippen springen. Alle Lemminge haben die Eigenschaft, dass sie das tun, was der Lemming vor ihnen tut. Kannst Du sicher sein, dass alle Lemminge im ersten Bild links über die Klippe springen? Überlege Dir, was bei den Induktionsschritten 1, 2 und 3 in diesem Beispiel zu notieren ist.

b. Warum springen die kanadischen Lemminge (im mittleren Bild) nicht? In den Sprechblasen steht jeweils «After you».

Lösung zur Aufgabe 2:

Verankerung: Der 1. Lemming springt.

Induktionsannahme: Der Lemming n «vor mir» springt.

Induktionsschritt $n \to n+1$: Jeder Lemming verhält sich gleich wie der vor ihm.

*Keiner der kanadischen Lemminge beginnt zu springen (die Verankerung fehlt also), ausserdem gibt es keine klare Ordnung von n zu $n+1$: Die Lemminge sind nicht nummeriert und somit gibt es keine klare Vorgänger-Nachfolger-Beziehung. D.h. der Induktionsschritt «**wenn** $A(n)$ gilt, **dann** $A(n+1)$» ist schlecht definiert.*

Die folgenden Aufgaben illustrieren, dass die Verankerung in einem Induktionsbeweis nicht immer schon mit $n = 0$ oder $n = 1$ anfängt.

<u>Erinnerung</u>: Die **Fakultät** einer Zahl $n \in \mathbb{N}$ wird durch das folgende Produkt definiert:

$$n! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot (n-1) \cdot n$$

Zusätzlich gilt $0! = 1$.

Aufgabe 3

Ist es für alle natürlichen Zahlen n korrekt, dass $\mathrm{n}! > 2^{\mathrm{n}}$? Finde heraus, ab welchem Wert von n dies gilt, und beweise danach das Resultat per vollständiger Induktion.

Lösung zur Aufgabe 3:

$$0! = 1 \ \ngtr 1 = 2^0$$

$$1! = 1 \ \ngtr 2 = 2^1$$

$$2! = 2 \ \ngtr 4 = 2^2$$

$$3! = 6 \ \ngtr 8 = 2^3$$

$$4! = 24 > 16 = 2^4$$

$$5! = 120 > 32 = 2^5$$

Es scheint also, dass $n! > 2^n$ für $n \geq 4$ gilt. Beweisen wir das mithilfe der Induktion.

Verankerung: $A(4)$: Für $n = 4$: $4! = 24 > 16 = 2^4$ ist korrekt.

Induktionsannahme (IA): $A(n)$*: Wir nehmen an, dass* $n! > 2^n$ *wahr ist für* $n \geq 4$.

Induktionsschritt $n \to n+1$*: Wir müssen zeigen, dass* $A(n+1)$ *stimmt:*

$$(n+1)! = \boldsymbol{n!} \cdot (n+1) \overset{IA}{>} \boldsymbol{2^n} \cdot (n+1) > 2^n \cdot 2 = 2^{n+1},$$

wobei $(n+1) > 2$ *gilt, weil* $n \geq 4$*. Somit gilt* $(n+1)! > 2^{n+1}$ *und* $A(n+1)$ *ist wahr.*

Aufgabe 4

Zeige mithilfe der vollständigen Induktion, dass jede Postgebühr ≥ 8 Eurocents kann ausschliesslich aus Briefmarken von 3 Eurocents und aus Briefmarken von 5 Eurocents zusammengestellt werden.

Tipp: Bei der Induktionsannahme $A(n)$ unterscheide 2 Fälle:

(1) Die Zusammenstellung aus n Cents enthält mindestens eine Briefmarke à 5 Cent und
(2) die Zusammenstellung aus n Cents enthält **keine** Briefmarke à 5 Cent.

Lösung zur Aufgabe 4:

Verankerung: $A(8)$*: 8= 3+5 ist korrekt.*

Induktionsannahme: $A(n)$*: Wir nehmen an, dass man dir Postgebühr von n Eurocents ausschließlich aus Briefmarken à 3 Eurocent und aus Briefmarken à 5 Eurocent zusammenstellen kann mit* $n \geq 8$.

Im 1. Fall enthält die Zusammenstellung aus n *Cents mindestens eine Briefmarke à* 5 *Cent.*

Induktionsschritt $n \to n+1$*: Wir ersetzen diese 5-Cent-Briefmarke durch zwei Briefmarken à* 3 *Cent. Somit wird die Postgebühr um 1 Cent erhöht, sie beträgt nun* $n+1$ *Cent und hat per Konstruktion Briefmarken von sowohl à 3 Cent als auch à* 5 *Cent.*

Im 2. Fall enthält die Zusammenstellung aus n *Cents keine Briefmarke à* 5 *Cent.*

Induktionsschritt $n \to n+1$*: Weil* $n \geq 8$ *und weil es keine 5 in* n *gibt, ist* $n = 9$ *das Minimum, d.h.,* n *enthält mindestens 3 Briefmarken à* 3 *Cent. Wir ersetzen diese drei Briefmarken à 3 Cent durch zwei Briefmarken à* 5 *Cent. Somit wird die Postgebühr um 1 Cent erhöht, sie beträgt also* $n+1$ *Cent und hat per Konstruktion Briefmarken sowohl à 3 als auch à* 5 *Cent.*

Somit ist $A(n+1)$ *in jedem Fall wahr.*

Die Methode der vollständigen Induktion ist aber nicht nur bei mathematischen Problemen einsetzbar, sondern auch in der Informatik, beispielweise um Algorithmen zu untersuchen. Wir illustrieren dies am Beispiel des Sortierverfahrens Bubblesort.

Aufgabe 5

a) Das Sortierverfahren ***Bubblesort*** kann man verwenden, um eine Liste (absteigend oder aufsteigend) zu sortieren. Es funktioniert so:
   1. Wir nehmen an, wir wollen die Eingabe $x_1, x_2, \ldots, x_{n-1}, x_n$ **aufsteigend** sortieren.

2. Die Eingabe wird von links nach rechts durchlaufen. Im ersten Schritt vergleichen wir das erste Element x_1 der Eingabe mit seinem rechten Nachbarn x_2.
3. Falls diese zwei Elemente «falsch geordnet» sind (d.h. für uns: falls $x_1 > x_2$), werden sie miteinander «in einer Bubble» vertauscht. Ansonsten verbleiben sie an ihren jeweiligen Stellen.
4. Danach geht man zu den Elementen x_2 und x_3 und wiederholt den Vergleich. Nur bei Bedarf (d.h. falls $x_2 > x_3$) werden sie getauscht, sonst nicht.
5. Man geht so vor bis zum Ende der Liste. Danach geht man zurück zum Beginn der Liste und iteriert den Prozess, bis die ganze Eingabe aufsteigend sortiert ist.

Nun bist Du dran! Wende Bubblesort an, um die Liste 1 = 5 7 1 2 aufsteigend zu sortieren.

Lösung 5a): *Rote Zahlen werden jeweils miteinander verglichen und bei Bedarf getauscht.*

5 7 1 2 → 7 **5 1** 2 → 7 1 **5 2** → 7 1 2 5 *(1. Durchlauf)*

7 1 2 5 → 1 **7 2** 5 → 1 2 **7 5** → 1 2 5 7 *(2. Durchlauf)*

b) Betrachte nun die Liste 2 = 9 1 2 5 7 . Was fällt Dir auf im Vergleich zur (sortierten) Liste 1? Wie viele Vertauschungen braucht man, um die Liste 2 mittels Bubblesort zu sortieren? Führe die Sortierung durch.

Lösung 5b): *Die Liste 2 ist gleich der sortierten Liste 1, mit einer 9 am Anfang. Man braucht 4 Vergleiche, um die Liste 2 zu sortieren.*

9 1 2 5 7 → 1 **9 2** 5 7 → 1 2 **9 5** 7 → 1 2 5 **9 7** → 1 2 5 7 9 *(1. Durchlauf)*

c) Und nun stellen wir uns eine allgemeinere Frage. Angenommen wir haben eine Liste n der Länge n, die bereits (aufsteigend) sortiert ist. Ein neues Element x kommt zu dieser Liste hinzu. Wie viele Vertauschungen müsste man **im schlimmsten Fall** mit Bubblesort durchführen, um die neue Liste n+1 der Länge $n + 1$ aufsteigend sortiert zu haben? Lass Dich von der Aufgabe b) inspirieren und versuche, eine induktive Formel für die schlimmste Anzahl Vertauschungen im allgemeinen Fall aufzustellen.

Lösung 5c): *Der allgemeine schlimmste Fall ist analog zum Fall in b). Das neue Element x müsste grösser sein als das Maximum in der sortierten Liste n , und x müsste am Anfang der Liste n, also links hinzugefügt werden. Dann benötigt man n Vertauschungen beim Bubblesort, bis x ganz rechts in der neuen Liste n+1 steht. Somit ist die schlimmste Anzahl Vertauschungen $V(n+1)$ in einer Liste der Länge n+1 im allgemeinen Fall um n grösser als die Anzahl Vertauschungen $V(n)$ in einer Liste der Länge n:*

$$V(n+1) = V(n) + n$$

d) Zeige mithilfe der vollständigen Induktion, dass die maximale Anzahl Vertauschungen in einer *beliebigen* Liste der Länge n, die man beim Bubblesort braucht, um die Liste aufsteigend zu sortieren, ist:

$$V(n) = \frac{n \cdot (n-1)}{2}$$

Lösung 5d):

Verankerung: Eine Liste der Länge $n = 1$ ist bereits sortiert. Die kleinste Liste, die man noch sortieren kann, hat Länge $n = 2$. Dabei ist $V(2) = 1$, weil eine Liste mit 2 Elementen im schlimmsten Fall umgekehrt geordnet ist und die zwei Elemente einmal umgetauscht werden müssen, damit ihre Reihenfolge stimmt.

Andererseits gilt die Formel $V(2) = \frac{2\cdot(2-1)}{2} = \frac{2\cdot(1)}{2} = 1$, *d.h., die Induktionsverankerung ist wahr.*

Induktionsannahme (IA): $V(n)$: *Wir nehmen an, dass* $V(n) = \frac{n\cdot(n-1)}{2}$ *wahr ist.*

Induktionsschritt $n \to n+1$: *Wir müssen zeigen, dass* $V(n+1)$ *auch wahr ist:*

$$V(n+1) = V(n) + n \overset{IA}{=} \frac{n\cdot(n-1)}{2} + n = \frac{n\cdot(n-1)+2\cdot n}{2} = \frac{n^2-n+2n}{2} = \frac{n^2+n}{2} = \frac{(n+1)\cdot n}{2}$$

Somit ist $V(n+1)$ *wahr.*

e) *Zusatzaufgabe für die Schnellsten*. Kannst Du die Formel $V(n) = \frac{n\cdot(n-1)}{2}$ für die maximal nötige Anzahl Vertauschungen in einer *beliebigen* Liste der Länge n direkt zeigen, und zwar ohne Anwendung des Induktionsverfahrens?

Lösung 5e): Im schlimmsten Fall ist die gegebene Liste am Anfang umgekehrt sortiert, also

$$x_n\,, x_{n-1}\,, \dots, x_2\,, x_1$$

mit $x_n > x_{n-1} > \cdots > x_2 > x_1$. *Dann benötigt man* $n-1$ *Vertauschungen, bis das grösste Element* x_n *ganz rechts ist:*

$$x_{n-1}\,, \dots, x_2\,, x_1, x_n$$

Danach benötigen wir nur noch $n - \mathbf{2}$ *Vertauschungen, bis das **zweit**grösste Element* x_{n-1} *an der vorletzten Stelle rechts ist:*

$$x_{n-2}\,, \dots, x_2\,, x_1, x_{n-1}, x_n$$

Das «$\boldsymbol{k}$-grösste» Element wird entsprechend nach $n - \boldsymbol{k}$ *Vertauschungen an der korrekten Stelle landen. Für die gesamte Liste muss man die Vertauschungen über alle* k *aufaddieren, um auf* $V(n)$ *zu kommen:*

$$\begin{aligned} V(n) = \sum_{k=1}^{n-1}(n-k) &= \sum_{k=1}^{n-1} n - \sum_{k=1}^{n-1} k = (n-1)\cdot n - \sum_{k=1}^{n-1} k \\ &= (n-1)\cdot n - [1+2+3+\cdots+(n-3)+(n-2)+(n-1)] \\ &= (n-1)\cdot n - \underbrace{[n+n+\cdots+n]}_{\frac{n-1}{2}\ \boldsymbol{Mal}} = (n-1)\cdot n - \frac{n\cdot(\boldsymbol{n}-\mathbf{1})}{\mathbf{2}} = \frac{n\cdot(n-1)}{2}. \end{aligned}$$

f) Erkläre mithilfe der folgenden Python-Implementierung der Methode `bubble_sort` und mithilfe der Teilaufgabe e), wie man sehen kann, dass die Worstcase-Laufzeit von Bubblesort in $O(n^2)$ für Eingaben der Länge n liegt.

```
3 def bubble_sort(array):
4     n = len(array)
5     for i in range(n-1):
6         for j in range(0, n-i-1):
7             if array[j] > array[j + 1] :
8                 (array[j], array[j + 1]) = (array[j + 1], array[j])
```

Lösung 5f): *Angenommen, dass die Laufzeit der Operationen «Variablenzuweisung» (Zeilen 4,8) und «Vergleich» (Zeile 7) je in $O(1)$ liegt, dann ist für die Berechnung der Laufzeit die Anzahl Durchläufe der beiden* `for`*-Schleifen relevant. In e) haben wir bewiesen, dass Bubblesort bei einer Eingabe der Länge n im schlimmsten Fall $\frac{n\cdot(n-1)}{2} \in O(n^2)$ Vertauschungen benötigt. Verglichen miteinander (Zeile 7) werden aber genau die Paare, die danach zu vertauschen sind (Zeile 8), d.h. das Verfahren braucht im schlimmsten Fall auch $\frac{n\cdot(n-1)}{2} \in O(n^2)$ Vergleiche. Insgesamt liegt also die Worstcase-Laufzeit von Bubblesort in $O(n^2)$ für Eingaben der Länge n.*

4. Fazit

Nach der Bearbeitung dieser Sequenz (ca. 1 Doppellektion für Aufgaben 1 bis 4 und ca. 1. Lektion (oder 1 Lektion + Rest als Hausaufgaben) für Aufgabe 5) haben die SuS die vollständige Induktion als Beweis- und Argumentationsmethode zum ersten Mal entdeckt. Sie erinnern sich an die 3 wichtigen Schritte des Induktionsbeweises und verstehen die Logik dahinter.

Die SuS sind sich auch bewusst, dass die Induktion nicht immer mit $n_0 = 0$ anfängt, sondern dass der Verankerungswert n_0 kontextabhängig zu wählen ist. Sie haben gesehen, dass die Induktionsbeweise nicht nur bei rein mathematischen Aussagen mächtig sind (Einführungsbeispiel, Aufgabe 3), sondern auch bei Rätseln (Aufgabe 4) oder bei der Untersuchung von Algorithmen (Aufgabe 5).