

Induktion: Anmerkungen für die Lehrperson

Zielsetzung

Das Ziel dieser Unterrichtseinheit ist die Einführung der vollständigen Induktion im Informatikunterricht. Dabei werden zwei Kontexte geboten: Exponentielles Wachstum und Sortieralgorithmen. Als Teilziele sollen die SchülerInnen (SuS)

- die Beweismethode vollständige Induktion kennenlernen,
- eine Vorstellung über exponentielles Wachstum entwickeln,
- die geometrische Summenformel für die Basis 2 kennenlernen,
- die diskrete Logarithmusfunktion kennenlernen,
- den Min-Sort Algorithmus kennenlernen,
- den Merge-Sort Algorithmus kennenlernen,
- erkennen, dass verschiedene Sortieralgorithmen unterschiedlich effizient sind.

Kontext Voraussetzungen

Die SuS befinden sich im fünften Schuljahr des Langzeitgymnasiums im Ergänzungsfach Informatik. Sie blicken zurück auf zwei Jahre Informatikunterricht.

- Aus der Mathematik kennen die SuS Potenzen mit natürlichen Exponenten.
- Aus der Mathematik und der Informatik haben die SuS Erfahrungen mit Zweierpotenzen gesammelt (Binärsystem).
- Aus der Informatik können die SuS in Python mit Funktionen, Schleifen, Verzweigungen und Listen umgehen.

Hingegen ist den SuS nicht bekannt,

- was ein Logarithmus sein soll,
- was eine Folge, das Summenzeichen oder eine Reihe ist,
- was ein Baum (im Kontext von Graphen) ist,
- wie man mit exponentiellem Wachstum umgeht.

Struktur

Die Unterrichtseinheit ist für etwa vier Lektionen ausgelegt. Die Ideen sind für die SuS vermutlich nicht einfach zu verstehen, weshalb die Lehrperson sie unterstützend begleitet.

Kapitel 1: Wachstum und Induktion

Zunächst wird im ersten Kapitel exponentielles Wachstum anhand von selbstreproduzierenden Robotern vorgestellt. Das Problem kann anhand von Darstellungen in der Klasse besprochen werden. Gemeinsam wird ein erster Induktionsbeweis diskutiert: die Anzahl Roboter in der h -ten Generation (Kapitel 1.1).

Als Rätsel sollen die SuS versuchen, selbst einen Induktionsbeweis durchzuführen. Das Problem (Kapitel 1.2) ist inhaltlich ähnlich wie das vorherige. Damit haben die SuS eine Chance, dieses zu knacken. Im Anschluss soll die Lösung besprochen werden. Als Fazit dient die geometrische Summenformel für die Basis 2.

Anhand des Wachstumsbeispiels soll der diskrete Logarithmus eingeführt werden: *Wie oft kann die Zahl durch 2 geteilt werden.* Die SuS sollen mit Python selbst eine Funktion programmieren, welche dies bewerkstelligt.

Zum Abschluss wird ein weiteres, klassisches Rätsel offeriert: Wie oft muss ein Stück Papier gefaltet werden, damit dieses zum Mond reicht.

Kapitel 2: Sortieralgorithmen

Anhand des gegebenen Pythoncodes sollen die SuS zuerst durchdenken, wie der Min-Sort Algorithmus funktioniert und die Funktionsweise in Worte fassen. Um die Anzahl Vergleiche zu berechnen, bietet sich hier ein erneuter Induktionsbeweis für die Gauss'sche Summenformel an.

In Kapitel 2.2 wird der Merge-Sort beschrieben. Der Merge-Sort soll als eine Anwendung der Induktion im Algorithmenentwurf dienen: Ausgehend von der Verankerung wird schrittweise eine grössere, sortierte Liste generiert. Voraussetzung ist, dass der Induktionsschritt gültig ist.

Zunächst soll anhand einer Abbildung und des Pythoncodes der Merge-Schritt nachvollzogen werden. Es wäre für die SuS zu schwierig, selbstständig zu programmieren.

Zum Schluss wird der Merge-Sort mit dem Min-Sort Algorithmus verglichen. Es wird zurückgegriffen auf die Logarithmusfunktion von Kapitel 1. Hier soll sich der Kreis schliessen.

Als zusätzliche Übung kann für beliebig grosse Listen die Sortierzeit gemessen und verglichen werden.

Es kann mit der Klasse diskutiert werden, dass der Merge-Sort Algorithmus zur Klasse der *Teile und Herrsche* Algorithmen gezählt wird. Teile und Herrsche ist eine Verallgemeinerung des Induktionsprinzips: Die Teilprobleme sind nicht notwendigerweise um genau eins kleiner wie bei der Induktion.