

## Didaktische Hinweise Rekursive Algorithmen

### Zielgruppe

Das niedersächsische Kerncurriculum Informatik für die gymnasiale Oberstufe<sup>1</sup> sieht das Konzept der Rekursion beim Entwurf von Algorithmen nur für Kurse auf erhöhtem Niveau vor. Die Materialien richten sich daher vorrangig an Schülerinnen und Schüler, die Informatik in der Qualifikationsphase auf erhöhtem Niveau belegt haben.

### Voraussetzungen

Die Erkundung rekursiver Algorithmen erfolgt in der Programmierumgebung Processing<sup>2</sup> auch unter Verwendung typischer Zeichenbefehle. Die Lernenden sollten bereits sicher in der Anwendung folgender algorithmischer Konzepte sein: Verwendung von elementaren Kontrollstrukturen, Variablen und statischen Reihungen sowie Erstellen eigener Methoden.

### Aufbau der Materialien und Lernziele

Das Konzept der Rekursion wird hier zunächst an rekursiv aufgebauten Grafiken eingeführt. Die entsprechenden Operationen benötigen keinen Rückgabewert und eine iterative Lösung äquivalenter Probleme ist den Lernenden vermutlich bereits bekannt. Der Fokus liegt daher zunächst auf der grundsätzlichen Idee, Wiederholungen durch Selbstaufruf, statt durch Schleifen zu realisieren.

Anhand mathematischer Berechnungen wird im zweiten Schritt der rekursive Ab- und Aufstieg bei rekursiv arbeitenden Operationen mit Rückgabewert thematisiert und veranschaulicht. In diesem Zusammenhang wird auch der erhöhte Speicherbedarf rekursiv arbeitender Algorithmen thematisiert. Die Aufgaben 6 und 8 nehmen außerdem praktische Grenzen bei der Berechnung durch den begrenzten Speicherplatz in den Blick. Während sich die Grenzen in Aufgabe 6 aus dem maximalen Speicherplatz für einen Wert abhängig vom Datentyp ergeben, ist in Aufgabe 8b die Begrenztheit der Anzahl der rekursiven Aufrufe, die im Hintergrund protokolliert werden können, die Ursache. Daran wird deutlich, dass eine iterative Lösung im Vergleich zu einer linearen Rekursion hinsichtlich des Speicherbedarfs häufig effizienter ist, auch wenn beide Lösungen logisch gleichwertig sind.

Abschließend wird das Prinzip *Teile und Herrsche* eingeführt, welches insbesondere bei der binären Suche zu einer deutlich schnelleren Lösung des Problems führt als eine lineare, iterative Suche. Zwar lässt sich auch eine binäre Suche iterativ implementieren, aber der rekursive Ansatz ist hier eine direkte Umsetzung des Prinzips *Teile und Herrsche* und daher oftmals intuitiver. Da die Anzahl der rekursiven Aufrufe in diesem Fall nur logarithmisch und nicht linear von der Anzahl der zu durchsuchenden Werte abhängt, ist der zusätzliche Speicherbedarf durch die Rekursion unproblematisch. Eine Gegenüberstellung der Laufzeit erfolgt im Rahmen des Materialpakets zur Komplexität von Algorithmen.

<sup>1</sup> Niedersächsisches Kultusministerium (Hrsg.) (2017) Kerncurriculum für das Gymnasium - gymnasiale Oberstufe, die Gesamtschule – gymnasiale Oberstufe, das Kolleg. Informatik. Hannover: unidruck

<sup>2</sup> Die Programmierumgebung Processing wurde 2001 von Ben Fry und Casey Reas initiiert. Nähere Informationen finden Sie unter <https://processing.org/>

Die Bearbeitung der Aufgaben des Materialpakets zur Rekursion fördert insbesondere die folgenden Kompetenzen<sup>3</sup>: Die Schülerinnen und Schüler ...

- erläutern das Konzept der Rekursion an gegebenen Beispielen
- entwerfen und implementieren rekursive Algorithmen
- erläutern die Strategie „Teile und Herrsche“ beim Entwurf rekursiver Algorithmen

### Ausblick

Die Probleme sind hier bewusst einfach gehalten, um das Konzept der Rekursion als alternativen algorithmischen Ansatz einzuführen, ohne die Lernenden gleich zu Beginn mit komplexen Problemen zu überfordern, die typischer Weise rekursiv gelöst werden, wie etwa das Problem der Türme von Hanoi. Solche Probleme können anschließend zur Vertiefung bearbeitet werden, wenn eine gewisse Sicherheit im Umgang mit rekursiven Algorithmen besteht.

Weiter vertieft werden kann das Konzept der Rekursion und des Prinzips *Teile und Herrsche* z. B. mit dem Sortieralgorithmus *MergeSort* (vgl. Materialpaket zur Komplexität von Algorithmen). Außerdem lernen die Schülerinnen und Schüler mit den Binärbäumen eine rekursiv aufgebaute Datenstruktur kennen, welche eine rekursive Problemlösung häufig nahelegt.

### Hinweis

Zu beachten ist, dass sich die Materialien zwar am niedersächsischen Kerncurriculum für die gymnasiale Oberstufe orientieren, jedoch keinen Anspruch auf Vollständigkeit hinsichtlich der für die Abiturprüfung erwarteten Kompetenzen erheben. Die Autorin hat zum Teil individuelle Schwerpunkte gesetzt, die auch über die im KC geforderten Kompetenzen hinausgehen können. Darstellungen und Schreibweisen können ggf. von den in der Abiturprüfung verwendeten Darstellungen und Schreibweisen abweichen. Verbindlich für das Abitur in Niedersachsen sind allein das niedersächsische Kerncurriculum für die gymnasiale Oberstufe (s. [1]) sowie die ergänzenden Hinweise (s. [2]) in der jeweils aktuellen Fassung. Es obliegt daher den jeweiligen Fachlehrkräften, den Unterricht so zu gestalten, dass die Schülerinnen und Schüler umfassend auf das Abitur vorbereitet werden. Die vorliegenden Materialien stellen somit nur eine Anregung dar, die an die individuellen Bedürfnisse der Lerngruppe angepasst werden müssen.

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht kommerziell - Keine Bearbeitungen 4.0 International Lizenz](#). Sie erlaubt Download und Weiterverteilung des vollständigen Werkes unter Nennung meines Namens, jedoch keinerlei Bearbeitung oder kommerzielle Nutzung.

Für die korrekte Ausführbarkeit der beiliegenden Quelltexte wird keine Garantie übernommen. Auch für Folgeschäden, die sich aus der Anwendung der Quelltexte oder durch eventuelle fehlerhafte Angaben ergeben, wird keine Haftung oder juristische Verantwortung übernommen.

---

<sup>3</sup> Niedersächsisches Kultusministerium (Hrsg.) (2017) Kerncurriculum für das Gymnasium - gymnasiale Oberstufe, die Gesamtschule – gymnasiale Oberstufe, das Kolleg. Informatik. Hannover: unidruck