

Didaktische Hinweise

Komplexität von Algorithmen

Zielgruppe

Das niedersächsische Kerncurriculum Informatik für die gymnasiale Oberstufe (s. [1]) sieht die Beurteilung der Effizienz 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 Lernenden sollten bereits sicher im Umgang mit algorithmischen Kontrollstrukturen sein und mit statischen Reihungen vertraut sein. Auch das Prinzip der Rekursion sollten die Lernenden beherrschen, um alle Aufgaben bearbeiten zu können.

In den Materialien werden insbesondere gegebene Such – und Sortiervverfahren systematisch analysiert und hinsichtlich ihrer Komplexität verglichen. Einen Such- und einen Sortieralgorithmus können die Lernenden aber grundsätzlich auch selbst entwickeln. Es sollte ihnen daher zunächst Gelegenheit gegeben werden, diese Probleme selbst zu lösen. Dazu können beispielsweise die Arbeitsblätter 01 und 02 aus dem Materialpaket „Suchen und Sortieren“ verwendet werden. Darauf aufbauend kann dann dieses Materialpaket zur Vertiefung und zur Thematisierung der Komplexität eingesetzt werden. Die Einstiegsaufgabe kann dabei als Überleitung dienen.

Die binäre Suche muss nicht zwingend vorher thematisiert werden, sondern kann hier im Laufe des ersten Teils zur Abschätzung der Laufzeit von Algorithmen erarbeitet werden.

Aufbau der Materialien und Lernziele

Mit den vorliegenden Materialien zur Komplexität von Algorithmen kann insbesondere die folgende Kompetenz aus dem Modul „Grundlagen der Algorithmik“ im Lernfeld „Algorithmen und Datenstrukturen“ gefördert werden (s. [1]):

Die Schülerinnen und Schüler beurteilen die Effizienz von Algorithmen unter der Abschätzung des Speicherbedarfs und der Zahl der Operationen.

01 Abschätzung der Laufzeit von Algorithmen

Der erste Teil führt die Lernenden in die Vorgehensweise bei der Abschätzung der Komplexität eines Algorithmus ein. Im Vordergrund steht dabei die Komplexität der Laufzeit. Dabei wird die Landau-Notation mit der Angabe der Komplexitätsklassen als $O(g(n))$ verwendet, da sie eine kompakte Darstellung und eine Abstraktion von der konkreten Laufzeit ermöglicht, auch wenn diese im niedersächsischen Kerncurriculum nicht explizit gefordert ist.

Als Beispiel für ein Problem mit unterschiedlich effizienten Lösungen werden hier zwei Algorithmen für eine lineare Suche und ein Algorithmus für eine binäre Suche untersucht und verglichen.

Als Variable für die Größe der Eingabe wird typischerweise der Buchstabe n verwendet. Es ist wichtig zu thematisieren, dass immer definiert sein muss, wofür n steht und das im Kontext auch die Verwendung anderer Bezeichnungen möglich ist. Im Zusammenhang mit Aufgabe 9 und 10 sollte auch deutlich werden, dass eine geschachtelte Schleife nicht zwingend in $O(n^2)$ liegt, sondern die

Laufzeit abhängig von der Anzahl der äußeren und inneren Schleifendurchläufe und von der Definition der Eingabegröße ist.

Aufgabe 11 dient als Ausblick, dass Sicherheit und Geheimhaltung häufig darauf basieren, dass Algorithmen zum „Knacken“ zu ineffizient sind und damit praktisch nicht umsetzbar. Während im Allgemeinen möglichst effiziente Algorithmen gesucht sind, wird hier ausgenutzt, dass es keinen effizienten Algorithmus gibt und das „Knacken“ daher an die Grenzen der praktischen Berechenbarkeit stößt.

Die Aufgaben üben und festigen das Thema nicht abschließend. Vielmehr sollten nach der Einführung im weiteren Verlauf des Unterrichts bei der Implementierung auch immer wieder Komplexitätsabschätzungen an geeigneter Stelle vorgenommen werden.

02 Sortialgorithmen im Vergleich

Der zweite Teil zum Vergleich der Sortialgorithmen kann je nach Unterrichtsverlauf direkt im Anschluss oder aber auch mit zeitlichem Abstand eingesetzt werden.

Für die verschiedenen Sortierv Verfahren bietet sich eine arbeitsteilige Erarbeitung an, da es vor allem um den Vergleich und weniger um das detaillierte Verständnis jedes einzelnen Algorithmus geht.

In der Lösung sind die Schritte der Sortialgorithmen tabellarisch visualisiert. Im Unterricht bietet sich alternativ ein enaktives Durchspielen der Algorithmen mit Karten oder ähnlichem an. Ggf. können den Lernenden auch die Tabellen aus der Lösung ohne Inhalt als Vorlage zum Ausfüllen zur Verfügung gestellt werden. Diese sind in der Datei *Tabelle_Vorlagen* im Lehrerordner enthalten.

In Aufgabe 7 erfolgt eine Implementierung der verschiedenen Sortialgorithmen, um die Laufzeiten auch praktisch zu vergleichen. Alternativ kann den Lernenden auch eine vollständige Implementierung zur Verfügung gestellt werden, wenn die Zeit für eine eigene Implementierung nicht ausreicht oder bereits an anderer Stelle erfolgt ist. Die Rechenzeiten können für die Sortierung von 500.000 oder 1.000.000 Werten mehrere Minuten bis zu einer halben Stunde beanspruchen. Das sollte bei der Planung berücksichtigt werden, so dass die Lernenden die Wartezeit für andere Aufgaben nutzen können und die Sortierung nicht vorzeitig abbrechen, da sie aufgrund der langen Laufzeit vielleicht einen Fehler vermuten. In dieser Aufgabe wird deutlich, dass die Einordnung in eine Komplexitätsklasse tatsächlich nur eine sehr grobe Abschätzung ist. Während das Wachstum der Laufzeit geeignet eingeordnet wird, können die konstanten Faktoren, die bei der Einordnung ignoriert werden, in der Praxis durchaus eine Rolle spielen.

Die Sortialgorithmen wurden in Processing implementiert. Eine Anpassung an andere Sprachen ist natürlich möglich. Es sollte nur zwecks Vergleichbarkeit innerhalb der Lerngruppe die gleiche Programmiersprache und Implementierung verwendet werden.

Anstatt die konkrete Messung von Laufzeiten am Ende durchzuführen, kann diese auch als Motivation zu Beginn einer Einheit zur Komplexität erfolgen. In diesem Fall können die Lernenden beispielsweise selbst entwickelte und implementierte Sortialgorithmen gegeneinander antreten lassen. Aufgabe 1 aus 01 Abschätzung der Laufzeit von Algorithmen kann dann basierend auf den Erfahrungen der Lernenden diskutiert werden.

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.

Literatur

- [1] Niedersächsisches Kultusministerium (Hrsg.) (2017) Kerncurriculum für das Gymnasium - gymnasiale Oberstufe, die Gesamtschule – gymnasiale Oberstufe, das Kolleg. Informatik. Hannover: unidruck
- [2] Niedersächsisches Kultusministerium (Hrsg.) (2021) Ergänzenden Hinweisen zum Kerncurriculum Informatik für die gymnasiale Oberstufe am Gymnasium und an der Gesamtschule sowie für das Kolleg. <https://cuvo.nibis.de/index.php?p=download&upload=260> [Datum des Zugriffs: 09.04.2025]
Hinweis: Ab dem Abitur 2027 gilt die Neufassung vom Juni 2025:
<https://cuvo.nibis.de/index.php?p=download&upload=736> [Datum des Zugriffs: 03.07.2025]

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.