

Fach Informatik - Grundkurs

Prüfender Fachlehrer Benjamin Julian Leinberger

Vorbereitungszeit: 20 min, Prüfungszeit 30 min

Sortieralgorithmen

1. Einordnung der Aufgabe in den Lehrplan, Taxonomie:

Im Lehrplan findet man das Thema Sortieralgorithmen in folgenden Lernbereichen:

1.1 Klassenstufe 8, Lernbereich 1: Algorithmen

1.1.1 Lehrplan

Lernbereich 1: Algorithmen		9 Ustd.
Kennen des Algorithmusbegriffes Eigenschaften Übertragen der Eigenschaften von Algorithmen auf Sachverhalte aus der Erfahrungswelt der Schüler Beherrschen der algorithmischen Lösung einfacher Problemstellungen in einer didaktisch reduzierten Programmierumgebung - Darstellungsformen von Algorithmen - Sequenz, Verzweigung und Wiederholung Kennen der Bedeutung von Algorithmen im gesellschaftlichen Kontext	 → Kl. 7, LB 3 → MA, Kl. 8, LB 3 Soziale Medien, Smarthome, In-App-Käufe, Werbung ⇒ Medienbildung Blocksprachen, visuelle Programmierumgebungen Blockdarstellung, verbale Beschreibung Zählschleife, kopfgesteuerte oder fußgesteuerte Schleife	

1.1.2 Lernzieltaxonomien (nach Anderson et al. (2001))

Stufe 1: Erinnern

Die SchülerInnen kennen den Algorithmus Begriff und dessen Eigenschaften

1.2 Klassenstufe 10, Lernbereich 1: Algorithmen

1.2.1 Lehrplan

Lernbereich 1: Algorithmen		11 Ustd.
Beherrschen der Implementierung der algorithmischen Grundstrukturen - Datentypen - Zahlen - Zeichenketten - Wahrheitswerte - Variablenzuweisungen - verknüpfte Bedingungen Kennen des Prinzips der Modularisierung Übertragen der Kenntnisse zu Algorithmen auf maschinelle Entscheidungsprozesse	grundlegende, einfache Algorithmen Syntax und Semantik → Kl. 8, LB 1 → LB 2 Verkettung durch logische Operatoren Nutzen von Unterprogrammen und Bibliotheken autonomes Fahren, Gesichtserkennung, Wahlcomputer → KL. 9, LB 2 → ETH, Kl. 10, LB 1	

1.2.2 Lernzieltaxonomien (nach Anderson et al. (2001))

Stufe 2: Verstehen

Die SchülerInnen beherrschen die Implementierung der algorithmischen Grundstrukturen.

1.3 Klassenstufe 10, Lernbereich 3: Komplexaufgabe

1.3.1 Lehrplan

Lernbereich 3: Komplexaufgabe		6 Ustd.
Gestalten eines Projektes zur Lösung einer informatischen Problemstellung	Verknüpfung von Lernbereichen der Klassenstufen 9 und 10 Erstellung und Testung einer KI Webanwendung Sortier- und Suchalgorithmen Backtracking Chat-Bot Erstellung eines Client-Server-Dienstes ⇒ Problemlösestrategien → KI, LB 3 → DE, KI, 9, LB 3	
Nutzung verschiedener Anwendungen	selbstständiges und planvolles Arbeiten an einer komplexen Aufgabenstellung in Teams ⇒ Empathie und Perspektivwechsel ⇒ Verantwortungsbereitschaft	

1.3.2 Lernzieltaxonomien (nach Anderson et al. (2001))

Stufe 3: Anwenden

Die SchülerInnen wenden Sortieralgorithmen an, um eine informatische Problemstellung zu lösen.

1.4 Klassenstufe 11/12, Lernbereich 2: Algorithmierung und Programmierung

1.4.1 Lehrplan

Lernbereich 2: Algorithmierung und Programmierung		20 Ustd.
<p>Kennen des erweiterten Algorithmusbegriffes</p> <p>Beherrschen der Implementierung strukturierter Datentypen in einer Programmierumgebung</p> <p>Beherrschen der Arbeit mit Unterprogrammen</p> <ul style="list-style-type: none"> - Struktur von Unterprogrammen - Verwendung von Parametern <p>Kennen von Konzepten der objektorientierten Programmierung</p> <ul style="list-style-type: none"> - Klasse und Objekt - Attribut und Attributwert - Methode <p>Übertragen des objektorientierten Paradigmas auf einfache Problemstellungen</p> <p>Kennen von Rekursion und Iteration</p> <p>Beherrschen der Implementierung eigener Lösungen zu ausgewählten komplexen Problemstellungen</p> <p>Kennen von Aspekten der Effizienz und Komplexität von Algorithmen</p> <p>Einblick gewinnen in die Grenzen der Berechenbarkeit</p>	<p>Eigenschaften von Algorithmen</p> <p>→ Kl. 8, LB 1 Zeichenkette, Feld</p> <p>→ Kl. 10, LB 1</p> <p>Vererbung, Kapselung, Polymorphie</p> <p>→ Kl. 7, LB 1 → Kl. 9, LB 1</p> <p>Fraktale</p> <p>Auswahl des Programmierparadigmas Spiel, Simulation, Sortieralgorithmen, Suchverfahren</p> <p>⇒ Problemlösestrategien</p> <p>Komplexitätsklassen, Zeitkomplexität, Speicherkomplexität</p> <p>Sortieralgorithmen, Problem des Handlungsreisenden, Vierfarbenproblem, Brückenproblem, Primfaktorzerlegung</p> <p>algorithmische Unlösbarkeit, Rucksackproblem, Hamiltonkreis, Halteproblem</p>	

1.4.2 Lernzieltaxonomien (nach Anderson et al. (2001))

Stufe 1: Erinnern

Die SchülerInnen kennen den erweiterten Algorithmus Begriff und dessen Eigenschaften.

Die SchülerInnen kennen Aspekte der Effizienz und Komplexität von Algorithmen

Stufe 2: Verstehen

Die SchülerInnen beherrschen die Implementierung eigener Lösungen zu ausgewählten komplexen Problemstellungen

2. **Aufgabenstellung (so wie sie dem Prüfling vorgelegt wird):**

Mündliche Prüfung Informatik: Sortieralgorithmen

Gegeben ist folgendes ARRAY von Werten die aufsteigend sortiert werden sollen.

	00	01	02	03	04	05	Index
Wert	4	1	9	3	7	10	Array

Aufgabe 1 (6 BE)

- Nennen Sie vier Eigenschaften die einen Algorithmus ausmachen.
- Erläutern Sie zwei dieser Eigenschaften.

Aufgabe 2 (15 BE)

- Nutzen Sie das oben gegebene Array um den Bubblesort- Algorithmus an der Tafel darzustellen. Das Array soll aufsteigend sortiert werden.
- Stellen Sie einen beliebigen weiteren Sortieralgorithmus an der Tafel vor, um das gleiche Array aufsteigend zu sortieren.
- Vergleichen Sie die beiden Sortieralgorithmen miteinander, hinsichtlich der Komplexität und der Arbeitsweise.

Aufgabe 3 (2 BE)

Beurteilen Sie ob Algorithmen in der heutigen Gesellschaft eine große Rolle spielen und warum.

Aufgabe 4 (2 BE)

Nehmen Sie Stellung zu der Aussage:

Ein Algorithmus ist das sinnvolle Durcheinander von Sequenzen, Schleifen und Verzweigungen - weiter nichts.

Aber diese können sehr aufwändig sein

4. Musterlösung mit Angabe der Zuordnung der einzelnen BE:

1.

a) Nennen Sie vier Eigenschaften die einen Algorithmus ausmachen. [Max. 4 BE]

- Ausführbarkeit
- Determinismus [1]
- Determiniertheit [1]
- Eindeutigkeit [1]
- Finitheit [1]
- Terminiertheit [1]

b) Erläutern Sie zwei dieser Eigenschaften [Max. 2 BE]

Ausführbarkeit [1]

Jeder Schrittmuss tatsächlich ausführbar sein.

Determinismus [1]

Zu jedem Zeitpunkt der Ausführung ist der Folgeschritt eindeutig bestimmt.

Determiniertheit [1]

Der Algorithmus muss unter gleichen Bedingungen stets das gleiche Ergebnis liefern.

Eindeutigkeit [1]

Ein Algorithmus darf keine widersprüchliche Beschreibung haben.

Finitheit [1]

Die Beschreibung des Algorithmus muss endlich sein.

Terminiertheit [1]

Der Algorithmus muss nach endlich vielen Schritten enden und ein Ergebnis liefern.

2.

a) Nutzen Sie das oben gegebene Array, um den Bubblesort- Algorithmus an der Tafel darzustellen. Das Array soll aufsteigend sortiert werden. [7]

		Bubblesort					
Vergleiche		00	01	02	03	04	05
0		4	1	9	3	7	10
1		4	1	9	3	7	10
2		1	4	9	3	7	10
3		1	4	9	3	7	10
4		1	4	3	9	7	10
5		1	4	3	7	9	10
6		1	4	3	7	9	10
7		1	4	3	7	9	10
8		1	3	4	7	9	10
9		1	3	4	7	9	10
10		1	3	4	7	9	10
11		1	3	4	7	9	10
12		1	3	4	7	9	10
13		1	3	4	7	9	10
14		1	3	4	7	9	10
--		1	3	4	7	9	10

- Vergleiche Kennzeichen [1]
- Tausch [1]
- Beschriftung [1]
- Sortierte Elemente bleiben bestehen [1]
- Ergebnis [1]

Erklärung

- Vergleich von Elementen, tausch falls nötig [1]
- Wiederholung bis vollständig [1]

b) Stellen Sie einen beliebigen weiteren Sortieralgorithmus an der Tafel vor, um das gleiche Array aufsteigend zu sortieren. [4]

Schritte	Selectionsort					
	0	1	2	3	4	5
0	4	1	9	3	7	10
1	4	1	9	3	7	10
2	4	1	9	3	7	10
3	4	1	9	3	7	10
4	4	1	9	3	7	10
5	4	1	9	3	7	10
6	1	4	9	3	7	10
7	1	4	9	3	7	10
8	1	4	9	3	7	10
9	1	4	9	3	7	10
10	1	4	9	3	4	10
11	1	3	9	4	4	10
12	1	3	9	4	4	10
13	1	3	9	4	7	10
14	1	3	9	4	7	10
15	1	3	4	9	7	10
16	1	3	4	9	7	10
17	1	3	4	9	7	10
18	1	3	4	7	9	10
19	1	3	4	7	9	10
20	1	3	4	7	9	10

Schritte	Sortesort					
	0	1	2	3	4	5
0	4	1	9	3	7	10
1	4	1	9	3	7	10
2	1	4	9	3	7	10
3	1	4	9	3	7	10
4	1	4	9	3	7	10
5	1	4	9	3	7	10
6	1	4	9	3	7	10
7	1	4	9	3	7	10
8	1	3	9	4	7	10
9	1	3	9	4	7	10
10	1	3	9	4	7	10
11	1	3	4	9	7	10
12	1	3	4	9	7	10
13	1	3	4	9	7	10
14	1	3	4	7	9	10
15	1	3	4	7	9	10
--	1	3	4	7	9	10

		Mergesort					
Schritte		0	1	2	3	4	5
		4	1	9	3	7	10
1		4	1	9	3	7	10
2		4	1	9	3	7	10
3		4	1	9	3	7	10
4		1	4	9	3	7	10
5		1	4	9	3	7	10
6		1	3	4	7	9	10
--		1	3	4	7	9	10

		Insertionsort					
Vergleiche		0	1	2	3	4	5
0		4	1	9	3	7	10
1		4	1	9	4	7	10
2		1	4	9	4	7	10
3		1	4	9	3	7	10
4		1	4	3	9	7	10
5		1	3	4	9	7	10
6		1	3	4	9	7	10
7		1	3	4	7	9	10
8		1	3	4	7	9	10
--		1	3	4	7	9	10

- Ausführung [3]
- Beschriftung [1]

c) Vergleichen Sie die beiden Sortieralgorithmen miteinander, hinsichtlich der Komplexität und der Arbeitsweise. [4]

- Komplexität [2]
- Arbeitsweise [2]

3. Beurteilen Sie ob Algorithmen in der heutigen Gesellschaft eine große Rolle spielen und warum. [2]

Ja Algorithmen spielen in unserer heutigen Gesellschaft eine große Rolle

- Algorithmen sind überall zu finden. [1]
- Digitalisierung aller gesellschaftlichen Bereiche [1]

4. Nehmen Sie Stellung zu der folgenden Aussage: [2]

Ein Algorithmus ist das sinnvolle Durcheinander von Sequenzen, Schleifen und Verzweigung- weiter nichts. Aber diese können sehr aufwändig sein!

- Zeitkomplexität, Mächtigkeit, Aufwand [1]
- alle Sortieralgorithmen nutzen mehr oder weniger tief geschachtelte Schleifen und verzweigen in Abhängigkeit des jeweiligen Ergebnisses oder eines Zwischenergebnisses [1]

5. Hinweise zur Umsetzung (benötigte Arbeitsmittel, ggf. Software auf dem Prüfungsrechner, ...):

Bei der Auswahl dieser Aufgabe ist zu beachten:

- Die Komplexität im Unterricht behandelt wurde
- Das eine Tafel und Kreide bzw. Tafelstifte im Prüfungsraum vorhanden sind.

6. Anhang: Abbildungen:

7. Quellenangabe, Abbildungsnachweise, ...:

<https://dbs-lin.ruhr-uni-bochum.de/lehreladen/wp-content/uploads/2022/07/verbliste-sammlung-deutsch.pdf>

(aufgerufen 05.03.2023)

<https://www.dr-datenschutz.de/was-ist-ein-algorithmus-definition-und-beispiele/>

(aufgerufen am, 05.03.2023)

8. Erklärung der Freigabe zur Nachnutzung der Aufgabe:

Hiermit erkläre ich Benjamin Julian Leinberger diese Aufgabe unter Wahrung des Urheberrechts erstellt zu haben.

Ich stelle diese Aufgabe zur Nachnutzung nach Lizenz CC BY-NC (Namensnennung, Bearbeitung, nicht kommerziell) zur Verfügung.



(Unterschrift des Autors / elektron. Signatur)