

Simulation einer mündlichen Abiturprüfung – Informatik

Thema: OOP, Algorithmen, Datenbanken, Formale Sprachen

May 31, 2025

Aufgabe 1: Analyse eines Algorithmus

Ein unbekannter Algorithmus zur Berechnung einer speziellen mathematischen Eigenschaft von Zahlen wird bereitgestellt. Der Name der Klasse lautet `Algorithmus`.

Gegebener Java-Code

```
public class Algorithmus {
    public static int berechne(int n) {
        if (n <= 1) {
            return n;
        }
        int a = 0, b = 1, c = 0;
        for (int i = 2; i <= n; i++) {
            c = a + b;
            a = b;
            b = c;
        }
        return c;
    }

    public static void main(String[] args) {
        int n = 5;
        System.out.println("Ergebnis: " + berechne(n));
    }
}
```

Teilfragen zur Analyse des Codes

1. Beschreiben Sie die Funktionsweise des Algorithmus. Welche mathematische Eigenschaft berechnet er?
2. Welche Art der Implementierung wurde hier verwendet und warum könnte sie vorteilhaft sein?

3. Verdeutlichen Sie die Funktionsweise des Algorithmus anhand eines Beispielaufrufs `berechne(5)`.
4. Wie viele Aufrufe werden insgesamt benötigt, um `berechne(5)` zu berechnen?
5. Bestimmen Sie die Zeitkomplexität des Algorithmus.
6. Wie verhält sich der Algorithmus im Vergleich zu einer intuitiven rekursiven Berechnung aller möglichen Werte?

Aufgabe 2: Normalisierung einer Datenbank für eine Schulbibliothek

Eine Schulbibliothek speichert Informationen zu ausgeliehenen Büchern in einer relationalen Datenbank. Die ursprüngliche Tabellenstruktur ist wie folgt:

Table 1: Ursprüngliche nicht normalisierte Tabelle

AusleiheID	SchülerID	Schülername	BuchID	Buchtitel	Autor	Datum
1	101	Max Meier	301	Mathematik für Einsteiger	A. Schmidt	01.03.20
2	102	Lisa Becker	302	Einführung in Java	B. Müller	02.03.20
3	101	Max Meier	303	Physik Grundlagen	C. Weber	03.03.20

Teilaufgaben zur Normalisierung

1. Identifizieren Sie Redundanzen in der Tabelle.
2. Zerlegen Sie die nicht normalisierte Tabelle und führen Sie eine schrittweise Normalisierung durch bis zur 3. Normalform.
3. Welche Vorteile bringt die Normalisierung in diesem Fall?

Aufgabe 3: Analyse einer formalen Grammatik

Eine Grammatik G erzeugt einfache arithmetische Ausdrücke mit Addition und Multiplikation:

- Terminale: $\{0, 1, +, *\}$
- Nichtterminale: $\{S, T, F\}$
- Startsymbol: S
- Produktionsregeln:

$$S \rightarrow S + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow 0 \mid 1$$

Teilfragen zur Grammatik

1. Leiten Sie den Ausdruck $1+1*0$ mit der Grammatik ab.
2. Welche Prioritätsregeln ergeben sich aus dieser Grammatik?
3. Wie könnte die Grammatik erweitert werden, um auch Klammern $()$ zu unterstützen?
4. Kann die Grammatik für alle möglichen mathematischen Ausdrücke erweitert werden? Begründen Sie Ihre Antwort.

Kolloquium

Algorithmen

1. Welche Eigenschaften muss ein Algorithmus haben?
2. Warum ist die iterative Lösung oft effizienter als eine rekursive Lösung?
3. In welchen Fällen könnte Rekursion einer Iteration vorzuziehen sein?
4. Was versteht man unter einem effizienten Algorithmus? Welche Maßstäbe werden zur Effizienzbewertung verwendet?
5. Wie analysiert man die Zeitkomplexität eines Algorithmus?
6. Warum ist $O(n \log(n))$ schneller als $O(n^2)$?
7. Was ist der Unterschied zwischen exponentiellen $O(2^n)$ und polynomiellen $O(n^k)$ Algorithmen?
8. Wie kann man den Speicherverbrauch eines Algorithmus reduzieren?
9. Schreiben Sie einen Algorithmus in Pseudo-Code, der zwei Variablen ohne eine zusätzliche Variable tauscht.
10. Können Sie eine weitere Möglichkeit das selbe Algorithmus mit Hilfe von booleschen Algebra zu implementieren?

Datenbanken

1. Warum ist die Normalisierung wichtig für eine relationale Datenbank?
2. Warum musste die ursprüngliche Tabelle normalisiert werden? Welche Probleme hätte es gegeben, wenn man sie in nicht normalisierter Form belassen hätte?
3. Gibt es in der endgültigen normalisierten Form noch Redundanzen? Falls ja, sind diese gewollt?
4. Welche konkreten Redundanzen wurden durch die Normalisierung beseitigt?

5. Welche Probleme können in einer nicht normalisierten Datenbank auftreten?
6. Gibt es Fälle, in denen man bewusst auf eine vollständige Normalisierung verzichtet? Warum?
7. Welche Normalform ist für den praktischen Einsatz am besten geeignet?
8. Schreiben Sie eine SQL-Abfrage, um ...

Sprachen und Grammatiken

1. Was versteht man unter einer formalen Sprache?
2. Was ist ein Alphabet?
3. Können Sie ein Beispiel für eine formale Sprache nennen?
4. Wie unterscheidet sich eine formale Sprache von einer natürlichen Sprache?
5. Was ist eine Grammatik, und aus welchen Komponenten besteht sie?
6. Erklären Sie den Unterschied zwischen Terminal- und Nichtterminalsymbolen.
7. Was ist der Unterschied zwischen einer regulären und einer kontextfreien Grammatik?
8. Was bedeutet eine kontextfreie Grammatik (CFG)? Geben Sie ein Beispiel.
9. Stellen Sie einen Vergleich zwischen Regulären und Kontextfreien Grammatik.
10. Sind Programmiersprachen wie beispielsweise Java oder C++ kontextfrei oder regulär.
11. Erstellen Sie eine kontextfreie Grammatik der Binärzahlen.

Eine kontextfreie Grammatik G erzeugt Binärzahlen:

- Terminale: $\{0, 1\}$
- Nichtterminale: $\{S\}$
- Startsymbol: S
- Produktionsregeln:

$$S \rightarrow 0S \mid 1S \mid 0 \mid 1$$