

Simulation einer mündlichen Abiturprüfung – Informatik

Thema: OOP, Algorithmen, Datenbanken, Formale Sprachen

March 30, 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;
        }
        return berechne(n - 1) + berechne(n - 2);
    }

    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 naiven iterativen Berechnung aller möglichen Werte?
7. Wie kann die Laufzeit verbessert werden? Welche alternative Implementierung könnte vorteilhafter sein?

Aufgabe 2: Datenmodellierung und Normalisierung

Eine Schule speichert die Noten der Schüler in einer relationalen Datenbank. Die ursprüngliche Tabellenstruktur ist wie folgt:

Table 1: Ursprüngliche nicht normalisierte Tabelle

SchülerID	Name	Klasse	Fach	Lehrer	Note
101	Max Meier	10A	Mathematik	Herr Schmidt	2
102	Lisa Becker	10A	Deutsch	Frau Müller	1
101	Max Meier	10A	Deutsch	Frau Müller	3
103	Anna Keller	10B	Englisch	Herr Weber	2
102	Lisa Becker	10A	Mathematik	Herr Schmidt	2

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

Gegeben sei die Grammatik G mit:

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

$$\begin{aligned}S &\rightarrow 0X1 \\X &\rightarrow 0X1 \mid 1Y0 \\Y &\rightarrow 1Y0 \mid \epsilon\end{aligned}$$

- Startsymbol: S

Überprüfen Sie, ob die folgenden Wörter zur Sprache $L(G)$ gehören:

- **Wort 1:** 000111000111
- **Wort 2:** 000100011000

Teilfragen zur Grammatik

1. Leiten Sie die Binärzahl 101 mit der Grammatik ab.
2. Kann die Grammatik alle möglichen Binärzahlen erzeugen? Falls nicht, wie könnte sie angepasst werden?
3. Wie könnte die Grammatik erweitert werden, um nur Binärzahlen mit gerader Länge zu erzeugen?

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?
Eine formale Sprache ist eine Menge von Wörtern über einem bestimmten Alphabet, die nach festen Regeln definiert sind.
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:

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