

Simulation einer mündlichen Abiturprüfung – Informatik

Thema: OOP, Algorithmen, Daten Banken, Formale Sprachen

April 28, 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 x, int y) {
        if (y == 0) {
            return x;
        }
        return berechne(y, x % y);
    }

    public static void main(String[] args) {
        int wert1 = 56;
        int wert2 = 98;
        System.out.println("Ergebnis: " + berechne(wert1, wert2));
    }
}
```

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(56, 98)`.
4. Wie viele Aufrufe werden insgesamt benötigt, um `berechne(56, 98)` 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?

Aufgabe 2: Datenmodellierung und Normalisierung

Ein Unternehmen verwaltet seine Kundenbestellungen und Produktinformationen in einer relationalen Datenbank. Die ursprüngliche Tabellenstruktur sieht wie folgt aus:

Table 1: Ursprüngliche nicht normalisierte Tabelle

BestellID	KundenID	Kundenname	Adresse	ProduktID	Produktname	Kategorie	Preis	Menge
1	1001	Anna Müller	Hauptstr. 12, Berlin	501	Laptop	Elektronik	1200	1
2	1002	Thomas Becker	Marktstr. 5, Hamburg	502	Smartphone	Elektronik	800	2
3	1001	Anna Müller	Hauptstr. 12, Berlin	503	Drucker	Bürobedarf	150	1
4	1003	Julia Schmitt	Lindenallee 9, München	504	Schreibtisch	Möbel	300	1
5	1002	Thomas Becker	Marktstr. 5, Hamburg	505	Monitor	Elektronik	200	1

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 ist gegeben durch:

- Terminale: $\{a, b\}$
- Nichtterminale: $\{S, A\}$
- Startsymbol: S
- Produktionsregeln:

$$\begin{aligned} S &\rightarrow aA \mid bS \mid a \mid b \\ A &\rightarrow aA \mid bA \mid a \end{aligned}$$

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

- **Wort 1:** aab
- **Wort 2:** $abba$

Teilfragen zur Analyse der Grammatik

1. Geben Sie die Produktionsschritte zur Ableitung von *aab* an, falls es zur Sprache gehört.
2. Zeigen Sie, warum *abba* nicht zur Sprache gehört, indem Sie eine Ableitung versuchen oder nachweisen, dass keine Ableitung möglich ist.
3. Beschreiben Sie in Worten, welche Struktur die von G erzeugten Wörter aufweisen.
4. Wie könnte die Grammatik verändert werden, um auch das Wort *abba* zu akzeptieren?

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 Nichtterminal- symbolen.
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:

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