

Algorithmen und Datenstrukturen

Name, Vorname:

Rekursion - Kontrollfragen

Mehrere Antworten können richtig sein

Frage 1. Warum braucht ein rekursives Programm oft mehr Speicher als eine entsprechende iterative Variante?

- ☐ Weil man jedes Zwischenergebnis abspeichern muss
- ☐ Weil die Anzahl rekursiver Aufrufe meistens grösser ist als die Anzahl Schleifendurchläufe im iterativen Algorithmus
- ☐ Weil die Anzahl rekursiver Aufrufe meistens grösser ist als die Anzahl Schleifendurchläufe im iterativen Algorithmus
- ☐ Weil der rekursive Algorithmus meistens eine grössere Laufzeit hat
- ☐ Weil bei jedem Methodenaufruf die Rücksprungsadresse gespeichert werden muss

Frage 2. Kann man jeden rekursiven Algorithmus so umbauen, dass er keine Rekursion verwendet?

- ☐ ja
- ☐ nein
- ☐ kommt drauf an

Frage 3. Was ist die wesentliche Idee für den rekursiven Algorithmus, um die Türme von Hanoi zu bewegen?

- ☐ Man teilt den Turm in zwei gleichgrosse Teile und ruft den Algorithmus für beide Teiltürme rekursiv auf
- ☐ Man ruft zunächst den Algorithmus für die obersten $n-1$ Scheiben auf und verschiebt diesen Teilturm, und bewegt anschliessend die unterste Scheibe auf einen anderen Stapel.
- ☐ Man bewegt zunächst die oberste Scheibe auf einen anderen Stapel und ruft dann den Algorithmus rekursiv für den verbleibenden Restturm auf

Frage 4. Sie finden folgendes Programm, um die Fakultät von n zu berechnen:

```
public int fac (int n) {  
    return n * fac(n-1);  
}
```

Was passiert, wenn Sie dieses Programm mit n=5 laufen lassen?

- ☐ Es bricht mit einer Fehlermeldung ab
- ☐ Es gibt 120 zurück
- ☐ Es gerät in eine Endlos-Schleife und läuft unendlich lange
- ☐ Etwas anderes

Frage 5. Was ist ein Indiz dafür, dass ein Programm rekursiv ist?

- ☐ Das Program läuft sehr schnell
- ☐ Das Programm ruft sich selber auf
- ☐ Das Programm verwendet eine Liste
- ☐ Das Programm läuft sehr langsam

Frage 6. Was haben Sie nicht verstanden? Falls alles klar war: Was fanden Sie am interessantesten?