

Hundert Fragen und Aufgaben zur Abiturvorbereitung Informatik

Zusammengestellt für das Abitur 2002

Die im Folgenden angegebenen Fragen und Aufgaben dienen als Leitlinien zur **Vorbereitung** auf die Abiturprüfung, insbesondere die mündliche Prüfung. Einige Fragen und Aufgabenstellungen erfordern kurze, die wichtigsten Informationen zusammenfassende Antworten, andere Fragestellungen sind so formuliert, dass sie eine längere, tiefergehende Auseinandersetzung erfordern. Sie können als offene Prüfungsfragen verstanden werden. Wichtige Grundbegriffe werden in Fettdruck angegeben.

I. Einführung in Algorithmen und Datenstrukturen

1. Was ist eine **Variable**, welche Komponenten gehören dazu?
2. Erläutern Sie, was man unter einem **Datentyp** versteht.
3. Erläutern Sie den Unterschied zwischen einer Variablen und einer **Datenstruktur**.
4. Geben Sie ein Beispiel für einen **Verbunddatentyp** (Record) in Ihrer Programmiersprache an.
5. Wodurch unterscheiden sich die Übergabearten **call by value** und **call by reference**.
6. Definieren Sie den Begriff **Algorithmus** und nennen Sie Synonyme.
7. Erläutern Sie den Unterschied zwischen einem Algorithmus und einem Programm.
8. Nennen und erläutern Sie wichtige **Eigenschaften von Algorithmen**.
9. Was beschreibt die **funktionale Spezifikation** eines Algorithmus?
10. Wann ist ein Algorithmus **total Korrekt**?
11. Welche Methoden zur Prüfung der Korrektheit eines Algorithmus sind Ihnen bekannt und wodurch unterscheiden Sie sich?
12. Die **Ulam-Folge** beginnt mit einer natürlichen Zahl a . Ist a gerade, so erhält man die nächste Zahl durch Halbierung; ist a ungerade, so erhält man die nächste Zahl dadurch, dass man a verdreifacht und dann 1 addiert. Die Folge endet, wenn ein Glied den Wert 1 hat. Bis heute ist es ein offenes Problem, ob die Ulam-Folge für jede natürliche Zahl endet. Was bedeutet dies für die Korrektheit eines eventuellen Algorithmus für die Ulam-Folge? Geben Sie einen (rekursiven) Algorithmus an.
13. Was ist eine **Iteration**?
14. Geben ein Beispiel für eine mathematische Operation an, die sowohl iterativ also auch rekursiv lösbar ist und geben Sie die jeweilige Vorschrift an.
15. Weshalb ist die **Abbruchbedingung** bei einer rekursiven Prozedur wichtig.
16. Für welche Probleme ist **Rekursion** als Lösungsverfahren sinnvoll?
17. Erläutern Sie den Begriff **Baumrekursion** und nennen Sie ein klassisches Beispiel.

18. Wovon ist die **Laufzeit** eines Algorithmus abhängig? Wovon ist die Laufzeit eines Programms abhängig?
19. Erläutern Sie grundsätzlich die Methode der Laufzeitanalyse nach einem **uniformen Komplexitätsmaß**.
20. Wie bestimmt man die Ordnung eines Algorithmus nach dem **O-Kalkül**?
21. Welche **Komplexitätsklassen** kennen Sie? Können Sie diese sinnvoll in eine Ordnung bringen?
22. Fassen Sie die **Gaußsche Summenformel**

$$f(n) = \frac{n \cdot (n + 1)}{2}$$

als Algorithmus auf und geben Sie eine Laufzeitabschätzung an.

23. Beschreiben Sie das Wachstumsverhalten der Gaußschen Summenformel in Abhängigkeit von n mit dem O-Kalkül.
24. Bestimmen Sie die Ordnung der folgenden Laufzeitfunktionen:

$$T(n) = 7n^5 + 3n^2 + 23$$

$$T(n) = 10n^2 + 5n^3 + 7n + 100$$

$$T(n) = 12$$

$$T(n) = 2^n + n^{10} + 70$$

25. Erläutern Sie den Standard-Sortieralgorithmus **Bubblesort** und geben Sie eine Laufzeitabschätzung an.

II. Konzepte der Abstraktion

1. Wie ist der Begriff **Datentyp** definiert?
2. Erläutern Sie, warum die Datenstruktur **Array** nur bedingt als Datentyp bezeichnet werden kann.
3. Was sind **strukturierte Datentypen**?
4. Was versteht man unter den Konzepten **Datenkapselung** und **Information Hiding**?
5. Was versteht man unter der **Schnittstelle** eines Datentyps?
6. Beschreiben Sie die Komponenten und Zustände einer Verkehrsampel, die bei der Modellierung eines entsprechenden Datentyps berücksichtigt werden müssen.
7. Wie kann eine Verkehrsampel als endlicher Automat modelliert werden? Zeigen Sie Gemeinsamkeiten in den Strukturen auf.
8. Geben Sie Eigenschaften und Operationen an, die zur sinnvollen Modellierung eines Karteikassensystems zum Sammeln *chemischer Fachbegriffe und ihrer Erläuterungen* notwendig sind.
9. Beschreiben Sie die Datenstruktur, die einem Telefonbuch unterliegt.
10. Erläutern Sie den Begriff **abstrakter Datentyp** allgemein.

11. Erläutern Sie den abstrakten Datentyp **Schlange** und geben Sie Anwendungsbeispiele an.
12. Geben Sie eine **Implementationsmöglichkeit** für eine Schlange an und erläutern Sie diese.
13. Erläutern Sie die Idee, die hinter der Implementation einer **Schlange als Ring** steht und geben Sie Vor- und Nachteile an.
14. Wodurch unterscheiden sich die abstrakten Datentypen **Keller** und **Schlange**?
15. Erläutern Sie den abstrakten Datentyp **Keller** und geben Sie Anwendungsbeispiele an.
16. Geben Sie eine **Implementationsmöglichkeit** für einen Keller an und erläutern Sie diese.
17. Betrachten Sie den Datentyp einer **sortierten Schlange**. Geben Sie einen Algorithmus an, der das Einfügen eines Eintrags in die Schlange an der richtigen Stelle ermöglicht.
18. Wie kann man eine Schlange mit zwei Kellern simulieren? Erläutern Sie die Idee!
19. Eine Datenstruktur zum Abspeichern von Zahlen zwischen 1 und 100 sei folgendermaßen definiert:
Es gibt zwei Behälter, einer mit der Aufschrift 1..50 und einer mit der Aufschrift 51..100. In dem Behälter mit der Aufschrift 1..50 stehen zwei weitere Behälter, einer mit der Aufschrift 1..25 und einer mit der Aufschrift 26..50, während im Behälter mit der Aufschrift 51..100 ebenfalls zwei weitere Behälter 51..75 und 75..100 stehen.
 - Welche Vorteile und Nachteile hat eine solche Datenstruktur gegenüber einem Behälter mit der Aufschrift 1..100?
 - Erläutern Sie, wie das Abspeichern einer Zahl und das Suchen einer Zahl in dieser Datenstruktur algorithmisch umgesetzt werden muß?
 - Wie könnte das Konzept erweitert werden, damit ein rekursiver Suchalgorithmus dabei Vorteile bringt?
20. Zwei sortierte Schlangen sollen zu einer dritten sortierten Schlange zusammengefügt werden. Erläutern Sie die Algorithmusidee.
21. Bei einem Taschenrechner in **umgekehrt polnischer Notation (UPN)** wird die Berechnung $(5 + 3) * 7 = 56$ folgendermaßen vorgenommen:
5 ENTER 3 ENTER + ENTER 7 ENTER * ENTER.
Erläutern Sie, welche Aufgabe ein Keller bei einem solchen Taschenrechner hat.
22. Warum ist Klammersetzung bei umgekehrt polnischer Notation nicht erforderlich?
23. Einige Implementationen der abstrakten Datenstrukturen *Schlange* und *Keller* sehen eine **Operation top** vor, die das erste/oberste Element der Datenstruktur liefert, ohne es von der Datenstruktur herunter zu nehmen. Wie könnte diese Operation bei einer Schlange/einem Keller durch Folgen anderer Operationen simuliert werden?
24. Was hat Rekursion mit der Datenstruktur Keller zu tun?
25. Der abstrakte Datentyp **sortierte Liste** soll sich von einer Schlange dadurch unterscheiden, dass Elemente auch aus dem Inneren der Liste herausnehmen und einfügen lassen. Machen Sie Vorschläge, wie die Operationen *get* und *insert* realisiert werden könnten. Was muss dabei beachtet werden?

III. Maschinennahe Konzepte

1. Aus welchen Komponenten besteht ein Computer?
2. Skizzieren und erläutern sie das **Von-Neumann-Konzept**
3. Was ist ein **universeller Rechner**? Welche anderen Konzepte gibt es noch?
4. Was ist das Besondere am Von-Neumann-Konzept bezüglich der Behandlung von **Programmcode** und **Daten**?
5. Was ist mit dem **Von-Neumann-Flaschenhals** gemeint?
6. Erläutern Sie das Grundkonzept der Programmiersprache **RePas**?
7. Welche **Adresssysteme** sind Ihnen bekannt? Welches wird bei der Programmiersprache RePas verwendet?
8. In der Programmiersprache RePas gibt es nur ein **Schleifenkonstrukt**? Welches ist das und warum genügt es, über nur ein Schleifenkonstrukt zu verfügen?
9. In der Programmiersprache RePas gibt es nur die Möglichkeit, Vergleiche mit 0 zu tätigen. Wie kann man sich behelfen, wenn man für zwei positive Integer- Zahlen a und b feststellen möchte, ob $a = b$ oder $a < b$ oder $a > b$ gilt?
10. Ersetzen Sie die folgende Pascal-For-Schleife durch eine RePas-Schleife.

```
for i:=1 to 10 do
  write(i);
```

11. Erläutern Sie das Konzept der Registermaschine.
12. Bei Registermaschinen gibt es einige ausgezeichnete Register, z.B. den **Akkumulator** und das **Befehlsregister**. Wozu dienen diese?
13. Ordnen Sie die **Strukturelemente des Von-Neuman-Konzeptes** vergleichbaren Elementen der **Registermaschine** zu und begründen Sie kurz.
14. Bei Registermaschinen gibt es verschiedene Adressierungsarten für den Zugriff auf Register, die mit $\#i$, i und $@i$ bezeichnet werden. Geben Sie diese an und erläutern Sie sie kurz.
15. Registermaschinen gehören zu den sogenannten **sequenziellen** Maschinenmodellen. Stellen Sie sich vor, eine Registermaschine wäre eine **Paralleltrechner** und könnte zwei oder mehrere Programme **parallel** ausführen. Welche Probleme würde dies mit sich bringen?
16. Definieren Sie einen **deterministischen endlichen Automaten**.
17. Definieren Sie einen **nichtdeterministischen endlichen Automaten**.
18. Worin besteht der Unterschied zwischen **Determinismus** und **Nichtdeterminismus** bei endlichen Automaten?
19. Geben Sie ein Beispiel für einen nichtdeterministischen endlichen Automaten an und überführen Sie ihn in einen deterministischen endlichen Automaten. Erläutern Sie dabei das Verfahren.

20. Welche Möglichkeiten gibt es, Nichtdeterminismus mit einem realen Rechner zu simulieren?
21. Wie kann ein endlicher Automat mit einem Pascal-Programm simuliert werden?
22. Geben Sie einen endlichen Automaten an, der die Wörter **begin** und **end** aus dem Alphabet $\Sigma = \{a, \dots, z, A, \dots, Z\}$ akzeptiert. Wie könnte der Automat zur Mustererkennung in einem Pascal-Quelltext eingesetzt werden?
23. Geben Sie einen deterministischen endlichen Automaten an, der einen Ticketautomaten simuliert: Ein Ticket soll 2,- Euro kosten. Der Automat erlaubt die Eingabe von 50-Cent-Stücken, 1-Euro und 2 Euro-Stücken.
24. Gegeben sei ein endlicher Automat $A = (\{0, 1\}, \{q_0, q_1, q_2, q_3\}, \delta, q_0, \{q_3\})$, wobei $\delta(q_0, 0) = \{q_0, q_1\}$, $\delta(q_0, 1) = \{q_0\}$, $\delta(q_1, 0) = \{q_2\}$, $\delta(q_1, 1) = \{q_2\}$, $\delta(q_2, 0) = \{q_3\}$, $\delta(q_2, 1) = \{q_3\}$. Zeichnen Sie den Graphen des Automaten und geben Sie einen äquivalenten, nichtdeterministischen endlichen Automaten an.
25. Erläutern Sie folgendes Beispiel eines regulären Ausdrucks $L(A) = (a|b)^*c^+def$ und geben Sie einen endlichen Automaten A an, der die Sprache akzeptiert.

IV. Sprachkonzepte - Robotik - Strukturierte Wiederholung

1. Erläutern Sie die Begriffe **Zeichen**, **Alphabet**, **Wörter** und **Sprache**
2. Wodurch unterscheidet sich eine **natürliche Sprache** von einer **formalen Sprache**?
3. Geben Sie eine formale Definition für eine Grammatik an und erläutern Sie diese.
4. Gegeben Sei eine Grammatik $G = (V, \Sigma, P, S)$ durch $V = \{S, A, B\}$, $\Sigma = \{a, b, c\}$ und $P = \{S \rightarrow AB, A \rightarrow ab, A \rightarrow aAb, B \rightarrow c, B \rightarrow cB\}$. Geben Sie eine Beispielerleitung für ein Wort der Sprache der Grammatik an.
5. Erläutern Sie die **Chomsky-Hierarchie**.
6. Auf welche Grundtechniken begründet sich die Robotik historisch betrachtet? Erläutern Sie diese kurz.
7. Welche Eigenschaften zeichnen einen autonomen, mobilen Roboter aus?
8. Erläutern Sie die Architektur der Kleinroboter, die im Unterricht eingesetzt wurden.
9. Nennen Sie beispielhaft einige Aspekte, durch die die Leistungsfähigkeit der im Unterricht eingesetzten Kleinroboter begrenzt ist.
10. Durch welche Konzepte ist Autonomie bei den im Unterricht verwendeten Kleinrobotern gewährleistet?
11. Erläutern Sie die Begriffe **Sensoren** und **Aktuatoren** und nenne Sie Beispiele.
12. Welche verschiedenen **Fortbewegungsmittel für mobile Roboter** kennen Sie und welche Vor- und Nachteile haben sie?
13. Erläutern Sie das Programmierungskonzept **Task** der **Programmiersprache NQC**.

14. Erläutern Sie kurz die Programmierprinzipien **Programmierung durch Beispiele**, **Programmierung durch Training**, **Roboterorientierte Programmierung** und **Aufgabenorientierte Programmierung**.
15. Welches ist das Grundprinzip der **verhaltensbasierten Programmierung** von Robotern?
16. Erläutern Sie den **Subsumtionsansatz von BROOKS** und geben Sie ein Beispiel für ein einfaches **Verhaltensnetz** an.
17. Wie kann die Umsetzung des Subsumtionsansatzes von BROOKS in der Programmiersprache NQC grundsätzlich geschehen?
18. Erläutern Sie das Konzept der **Programmierung dynamischer Prozesse** nach STEELS.
19. Welche verschiedenen Typen von **Quantitäten** unterscheidet man beim Konzept von STEELS?
20. Welche Schritte gehören zu einem **Zyklus** beim Konzept von STEELS?
21. Mit welchem Sprachkonstrukt der Sprache NQC können die **Quantitäten** nach STEELS umgesetzt werden? Warum entfallen dann die Methoden *AddValue(Quantität, Wert)* und *Value(Quantität)*?
22. Worauf muß man bei der Programmierung von Quantitäten achten, wenn man die Programmiersprache NQC verwendet?
23. Was versteht man unter **Embodiment**?
24. Im Zusammenhang mit paralleler Prozessverarbeitung ist häufig von **Quasiparallelverarbeitung** die Rede. Was ist damit genau gemeint?
25. In welchem Zusammenhang tritt der Begriff **Multitasking** noch auf und was ist damit gemeint?