



Teil 1: Beispielaufgaben

Die Illustrierenden Prüfungsaufgaben (Teil 1: Beispielaufgaben, Teil 2: Erläuterungen und Lösungsvorschläge) dienen der einmaligen exemplarischen Veranschaulichung von Struktur, Anspruch und Niveau der Abiturprüfung auf grundlegendem bzw. erhöhtem Anforderungsniveau im neunjährigen Gymnasium in Bayern.

Informatik

erhöhtes Anforderungsniveau

Arbeitszeit: 300 Minuten

Bei der Bearbeitung der Aufgaben dürfen folgende Hilfsmittel verwendet werden:

- das vom Staatsministerium genehmigte Dokument mit mathematischen Formeln
- ein Taschenrechner, der den vom Staatsministerium getroffenen Regelungen entspricht

Es werden vier Aufgaben zur Auswahl gestellt, von denen drei bearbeitet werden müssen.

Das Geheft mit den Aufgabenstellungen ist abzugeben.

1 Das P-Seminar INF26 hat sich zur Aufgabe gemacht, einen einfachen Vokabeltrainer zu entwickeln, der über einen Lexikonmodus und über einen Übungsmodus verfügen soll.

- a Zunächst muss entschieden werden, ob bei der Softwareentwicklung nach dem Wasserfallmodell oder nach einem agilen Modell vorgegangen werden soll. Beschreiben Sie drei grundlegende Unterschiede zwischen diesen Vorgehensmodellen.

3

An das Programm werden u. a. folgende Anforderungen gestellt:

- Der Lexikonmodus dient zum Suchen, Anzeigen und Bearbeiten der Vokabeleinträge.
- Im Übungsmodus soll der Nutzer zu einem vom Programm ausgewählten deutschen Wort die englische Übersetzung eingeben, wobei vereinfachend davon ausgegangen wird, dass genau eine Übersetzung richtig ist. Das Programm gibt anschließend Rückmeldung darüber, ob die Eingabe die erwartete englische Übersetzung ist.
- Es soll im Übungsmodus möglich sein, eine Jahrgangsstufe und eine Lektion auszuwählen, um nur die zu dieser Jahrgangsstufe und zu dieser Lektion gehörenden neuen Vokabeln üben zu können.

- Zudem soll im Übungsmodus gespeichert werden, wie oft ein deutsches Wort aufgerufen und wie oft eine falsche Übersetzung eingegeben wurde.

Für die Umsetzung vereinbart das P-Seminar Folgendes:

- Die aus der Abbildung ersichtlichen Daten sowie die Anzahl der Aufrufe und die Anzahl der falschen Eingaben werden in Objekten der Klasse VOKABEL gespeichert.
- Zur Verwaltung der Vokabel-Objekte soll ein nach den deutschen Wörtern alphabetisch geordneter Binärbaum zum Einsatz kommen.

- b Ein Schüler des P-Seminars behauptet, dass im Lexikonmodus die Suche nach einer bestimmten Vokabel bei einer sehr großen Vokabelanzahl in einem balancierten geordneten Binärbaum deutlich schneller gelingt als in einer verketteten Liste.

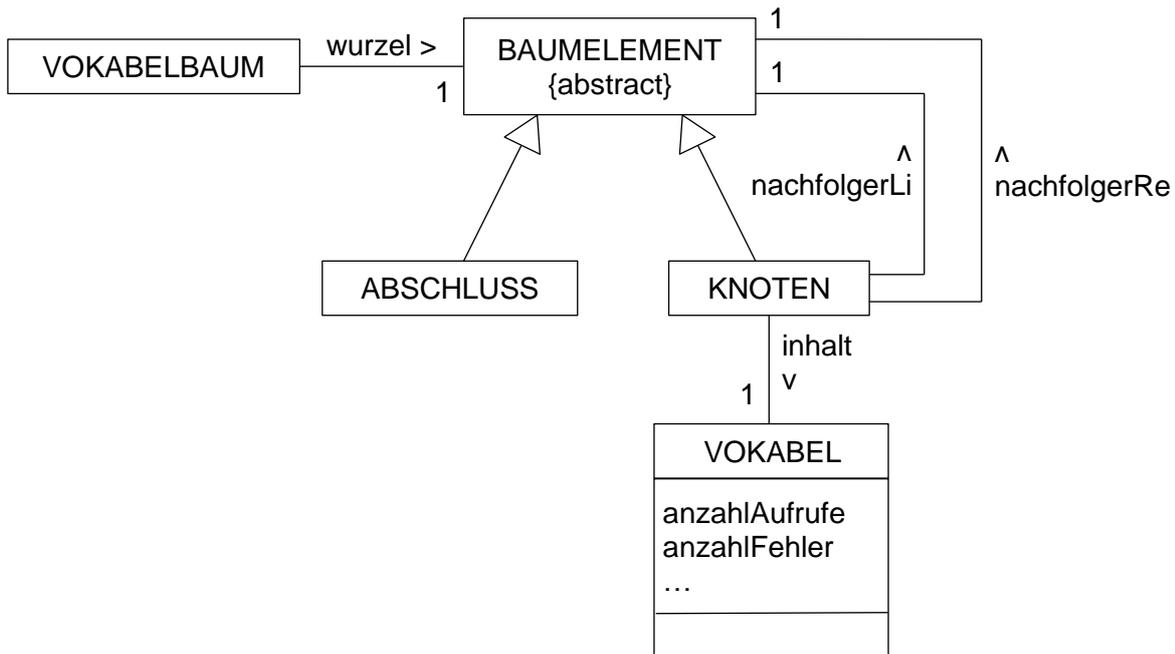
3

Beschreiben Sie, was unter einem balancierten Binärbaum zu verstehen ist.

Überprüfen Sie zudem die aufgestellte Behauptung an einem Beispiel mit 3500 Vokabeln.

(Fortsetzung nächste Seite)

Dem geordneten Binärbaum wird folgendes Klassendiagramm zugrunde gelegt:



c Geben Sie mithilfe einer auf dem Deckblatt angegebenen Programmiersprache eine Implementierung einer Methode *pruefen(eingabe)* in der Klasse VOKABEL an, die genau dann *wahr* zurückgibt, wenn die übergebene Zeichenkette gleich der erwarteten englischen Übersetzung ist. Die Methode soll außerdem die Anzahl der Aufrufe und ggf. die Anzahl der falschen Eingaben geeignet anpassen.

3

In der Klasse VOKABEL darf eine Methode *istRichtig(eingabe)* vorausgesetzt werden, die genau dann *wahr* zurückgibt, wenn die übergebene Zeichenkette mit der englischen Übersetzung übereinstimmt.

d In den geordneten Binärbaum sollen Objekte der Klasse VOKABEL eingefügt werden können.

7

Implementieren Sie mithilfe einer auf dem Deckblatt angegebenen Programmiersprache für alle Klassen der Baumstruktur die dazu notwendigen Methoden. Falls bereits ein Vokabel-Objekt für das deutsche Wort der einzufügenden Vokabel im Baum vorhanden ist, soll nichts eingefügt werden.

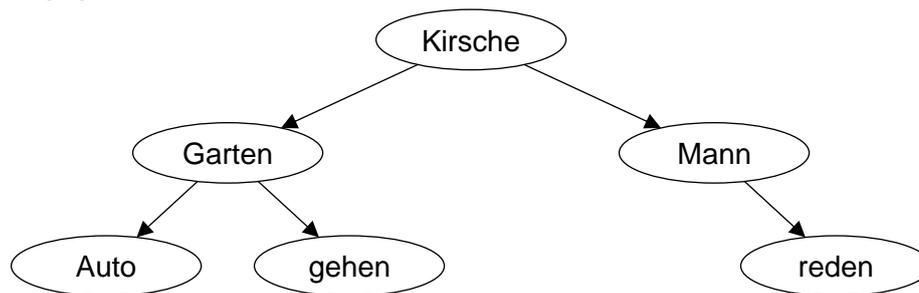
Folgende Methoden der Klasse VOKABEL dürfen als bereits implementiert vorausgesetzt werden:

- *istKleiner(andereVokabel)* gibt genau dann *wahr* zurück, wenn das deutsche Wort des ausführenden Objekts alphabetisch vor dem des übergebenen Objekts steht.
- *istGleich(andereVokabel)* gibt genau dann *wahr* zurück, wenn das deutsche Wort des ausführenden Objekts und des übergebenen Objekts übereinstimmen.

(Fortsetzung nächste Seite)

- e Beim nachstehenden geordneten Binärbaum sind nur die Schlüsselwerte, also die deutschen Wörter, angegeben.

2



Geben Sie die Reihenfolge der Schlüsselwerte an, wenn man den Baum mithilfe der Preorder-Traversierung durchläuft.

Geben Sie außerdem ein deutsches Wort an, das beim Einfügen in diesen Baum die Anzahl der Ebenen nicht verändert.

- f Zum Üben aller Vokabeln soll aus dem gesamten Binärbaum wiederholt eine Vokabel zufällig ausgewählt werden können. Dazu wird ein Algorithmus vorgeschlagen, für dessen Umsetzung folgende Methoden ausschlaggebend sind.

2

In der Klasse KNOTEN:

```
Methode auswaehlen()  
  wenn zufallszahl(0, 1) ist gleich 0 dann  
    vokabel = nachfolgerLi.auswaehlen()  
  sonst  
    vokabel = nachfolgerR.auswaehlen()  
  endeWenn  
  wenn vokabel ist leere Referenz dann  
    gib inhalt zurück  
  sonst  
    gib vokabel zurück  
  endeWenn  
endeMethode
```

In der Klasse ABSCHLUSS:

```
Methode auswaehlen()  
  gib leere Referenz zurück  
endeMethode
```

Der Aufruf *zufallszahl(0, 1)* liefert zufällig eine der Zahlen 0 oder 1.

Begründen Sie z. B. anhand des Baums aus Teilaufgabe 1e, warum der Algorithmus nicht dazu geeignet ist, alle Vokabeln gleich häufig zu üben.

(Fortsetzung nächste Seite)

Es soll möglich sein, aus dem geordneten Binärbaum diejenigen Vokabel-Objekte, die zu einer vorgegebenen Jahrgangsstufe und Lektion gehören, herauszufiltern und alphabetisch sortiert in einer Liste zu speichern. Zu diesem Zweck soll die Klasse VOKABELBAUM eine Methode *listeErstellen* besitzen, der als Parameter unter anderem eine Referenz auf eine zunächst leere Liste übergeben wird.

g Formulieren Sie, z. B. in Pseudocode, einen Algorithmus einer Methode *listeErstellen* der Klasse KNOTEN, die von der Methode *listeErstellen* der Klasse VOKABELBAUM verwendet werden kann. Dabei können Sie auf eine Methode *hintenEinfuegen(vokabel)* der Klasse LISTE zurückgreifen, die ein Objekt der Klasse VOKABEL am Ende der Liste einfügt.

4

Die Klasse VOKABELTRAINER verwaltet den oben betrachteten Binärbaum, der alle Vokabeln enthält, sowie eine Liste der aktuell zu übenden Vokabeln jeweils in einem Referenzattribut. Sie können im Folgenden davon ausgehen, dass die Klasse VOKABELBAUM einen Konstruktor zum Erzeugen eines leeren Baums und die Klasse LISTE einen Konstruktor zum Erzeugen einer leeren Liste zur Verfügung stellt.

h Geben Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine Implementierung der Klasse VOKABELTRAINER an, sodass ein Vokabeltrainer-Objekt erzeugt werden kann, das einen leeren Vokabelbaum sowie eine leere Liste referenziert.

2

i In der Klasse VOKABELTRAINER soll eine Methode *aktuelleListeErstellen(jahrgangsstufe, lektion)* zur Verfügung gestellt werden, die dafür sorgt, dass die Liste der aktuell zu übenden Vokabeln genau die zu den Parametern *jahrgangsstufe* und *lektion* passenden Vokabeln enthält. Stellen Sie kurz dar, welche Schritte in der Methode *aktuelleListeErstellen* durchgeführt werden müssen.

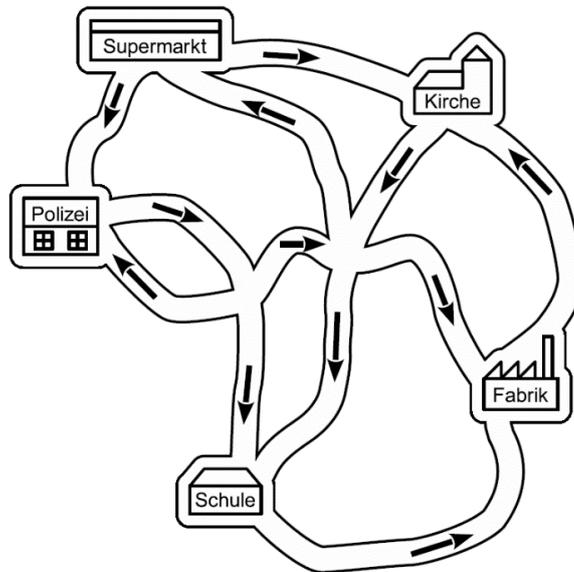
2

j Um die Suche zu beschleunigen, schlägt eine Schülerin vor, die Anzahl der Ebenen im Suchbaum zu reduzieren, indem jeder der Knoten bis zu fünf Nachfolger erhält. Geben Sie zwei Argumente an, die gegen die vorgeschlagene Änderung sprechen.

2

(Fortsetzung nächste Seite)

- 2 Das kleine Dorf Infonia soll verkehrsberuhigt werden. Dazu werden alle Straßen zu Einbahnstraßen umgewidmet (siehe Abbildung).



Die verkehrsberuhigende Maßnahme ist nur sinnvoll, wenn weiterhin jeder Ort im Dorf (Supermarkt, Kirche, Fabrik, Schule, Polizei, Kreuzung, ...) von jedem anderen Ort aus erreichbar ist. Um dies zu überprüfen, wird das Einbahnstraßennetz in einem Graph modelliert, dessen Knoten für die Orte und dessen Kanten für die jeweiligen Einbahnstraßen stehen.

Die Klasse GRAPH hat als Attribute u. a. ein zweidimensionales Feld *matrix* für die Adjazenzmatrix und ein eindimensionales Feld *knoten* mit Referenzen auf Objekte der Klasse KNOTEN. Die Methode *indexGeben* der Klasse GRAPH gibt den Feldindex zum als Parameter übergebenen Knoten zurück. Die Klasse KNOTEN hat ein Attribut *markierung* zur Markierung eines Knotens und verfügt über die beiden Methoden *markierungSetzen* und *markierungGeben*.

- a Nun soll für die Klasse GRAPH eine Methode *testen* entwickelt werden, die die geforderte Erreichbarkeit überprüft und eine entsprechende Rückmeldung liefert. Geben Sie eine mögliche Implementierung dieser und ggf. weiterer dafür nötigen Methoden in einer auf dem Deckblatt angegebenen Programmiersprache an. 9
- b Entscheiden Sie begründet, ob ein ungerichteter Graph mit denselben oben genannten Knoten das gegebene Straßennetz vollständig hätte repräsentieren können, bevor die Straßen zu Einbahnstraßen umgewidmet wurden. 1

1 Es wird eine Registermaschine mit folgendem Befehlssatz verwendet:

- load x kopiert den Wert aus der Speicherzelle x in den Akkumulator
- loadi n lädt die ganze Zahl n in den Akkumulator
- store x kopiert den Wert aus dem Akkumulator in die Speicherzelle x
- add x addiert den Wert aus der Speicherzelle x zum Wert im Akkumulator
- addi n addiert die ganze Zahl n zum Wert im Akkumulator
- sub x subtrahiert den Wert aus der Speicherzelle x vom Wert im Akkumulator
- subi n subtrahiert die ganze Zahl n vom Wert im Akkumulator
- mul x multipliziert den Wert im Akkumulator mit dem Wert in Speicherzelle x
- muli n multipliziert den Wert im Akkumulator mit der ganzen Zahl n
- div x dividiert den Wert im Akkumulator durch den Wert in Speicherzelle x (ganzzahlige Division)
- divi n dividiert den Wert im Akkumulator durch die ganze Zahl n (ganzzahlige Division)
- jmp x springt zum Befehl in der Speicherzelle x
- jmpn x springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator negativ ist
- jmpz x springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator gleich 0 ist
- jmpp x springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator positiv ist
- jmpnn x springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator nicht negativ ist
- jmpnz x springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator ungleich 0 ist
- jmpnp x springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator nicht positiv ist
- hold beendet die Abarbeitung des Programms

Zur Umwandlung einer nichtnegativen ganzen Zahl von der Dezimaldarstellung in die Binärdarstellung kann folgender Algorithmus angewendet werden, wobei *zahlDez* als Parameter für die einzugebende Zahl in Dezimaldarstellung steht. Nach Abarbeitung des Algorithmus steht in der Variablen *zahlBin* die umgewandelte Zahl, deren Darstellung als Binärzahl interpretiert werden kann.

rest = 0
zahlBin = 0
stelle = 1
wiederhole solange zahlDez > 0
rest = zahlDez mod 2
zahlBin = zahlBin + stelle * rest
zahlDez = zahlDez / 2
stelle = stelle * 10

Hinweis: Der Operator „/“ liefert den ganzzahligen Wert des Quotienten (ohne Rest) und der Operator „mod“ den Rest. Beispielsweise gilt: $7 / 2 = 3$ und $7 \text{ mod } 2 = 1$.

a Zeigen Sie anhand einer Tabelle für die Eingabe *zahlDez* = 13, welche Werte die Variablen *zahlDez*, *rest*, *zahlBin* und *stelle* im Laufe der Abarbeitung des Algorithmus annehmen.

2

(Fortsetzung nächste Seite)

- b** Begründen Sie, dass die Anweisung $rest = zahlDez \bmod 2$ des obigen Algorithmus mit der gegebenen Registermaschine durch die folgende Sequenz umgesetzt werden kann. 2
- ```

load zahlDez
divi 2
muli 2
sub zahlDez
muli -1
store rest

```
- c** Übertragen Sie den gegebenen Algorithmus zur Umwandlung einer nichtnegativen ganzen Zahl von der Dezimaldarstellung in die Binärdarstellung in ein Programm für die gegebene Registermaschine. Verwenden Sie symbolische Adressierung der Speicherzellen und Sprungmarken. 4
- 2** Eine Möglichkeit zur Darstellung von positiven und negativen Binärzahlen ist die Darstellung im Zweierkomplement.
- a** Erklären Sie anhand der beiden 8-Bit-Zahlen  $00010110_2$  und  $10100010_2$ , woran man erkennt, ob es sich um eine negative oder um eine positive Zahl handelt, und geben Sie für beide Zahlen die Dezimaldarstellung an. 2
- b** Geben Sie die größte sowie die kleinste 16-Bit-Zahl im Zweierkomplement an. Bestimmen Sie zudem den Zahlenbereich in Dezimalschreibweise, der mit einer 16-Bit-Zahl im Zweierkomplement dargestellt werden kann. 1
- c** Addieren Sie folgende 8-Bit- bzw. 4-Bit-Zahlen und geben Sie das Ergebnis im Zweierkomplement sowie als Dezimalzahl an. 4
- i**  $10110101_2 + 11111101_2$
- ii**  $1101_2 + 1010_2$
- Erläutern Sie anhand der Beispiele auch die Begriffe Übertrag und Überlauf und erklären Sie, woran man bei einer Addition im Zweierkomplement erkennt, dass ein Überlauf stattgefunden hat.
- 3** Eine Jury aus drei Personen entscheidet, ob sich ein Kandidat bzw. eine Kandidatin für die Finalrunde eines Wettbewerbs qualifiziert. Dazu müssen mindestens zwei der drei Jurymitglieder für den Kandidaten bzw. die Kandidatin stimmen. Wenn der Kandidat bzw. die Kandidatin die Finalrunde erreicht, soll eine grüne Lampe aufleuchten, ansonsten eine rote.
- Dies wird mit einem Schaltnetz umgesetzt, wobei die Stimme eines Jurymitglieds als boolesche Eingabe  $x$ ,  $y$  bzw.  $z$  realisiert wird.
- a** Stellen Sie die zugehörige Schalttabelle auf und leiten Sie daraus den Schaltterm für die grüne Lampe in disjunktiver Normalenform ab. 2
- b** Zeichnen Sie das Schaltnetz für die rote Lampe und erklären Sie kurz die verwendeten Logikgatter. 4
- c** Erläutern Sie, was durch den Schaltterm  $x \wedge y \vee x \wedge z \vee y \wedge z$  im Sachzusammenhang beschrieben wird. 1

(Fortsetzung nächste Seite)

- 4 In einem Betriebssystem für einen Prozessor werden folgende Prozesse betrachtet, wobei die Zeiten in Millisekunden und die Prioritäten im Bereich von 0 bis 2 gegeben sind. Die höchste Priorität wird mit 0 gekennzeichnet, die niedrigste mit 2.

| Prozess | Ankunftszeitpunkt | Laufzeit | Priorität |
|---------|-------------------|----------|-----------|
| P1      | 0                 | 4        | 1         |
| P2      | 0                 | 6        | 0         |
| P3      | 1                 | 1        | 1         |
| P4      | 3                 | 5        | 1         |
| P5      | 4                 | 2        | 2         |
| P6      | 4                 | 3        | 2         |
| P7      | 7                 | 2        | 0         |
| P8      | 9                 | 4        | 0         |

Zum Ankunftszeitpunkt wird ein Prozess in die Warteschlange eingefügt, sofern nicht der Prozessor gerade frei und der neu ankommende Prozess sofort zur Ausführung ausgewählt wird. Wird ein Prozess zum Zeitpunkt  $x$  unterbrochen, so reiht er sich mit Ankunftszeitpunkt  $x$  wieder in die Warteschlange ein. Sollen zwei Prozesse zum gleichen Zeitpunkt in die Warteschlange eingereiht werden, erfolgt die Sortierung nach ihren Namen (P1 vor P2 etc.).

- a Zwei bekannte Schedulingstrategien sind *First Come First Serve* sowie *Round Robin*. Erläutern Sie die Grundzüge beider Strategien und stellen Sie dabei die wesentlichen Unterschiede zwischen den beiden Strategien heraus. 4
- b Geben Sie für die Strategien *Shortest Job First* und *Round Robin* mit einem Zeitquantum  $q = 3$  sowie für ein selbstgewähltes Prioritätsschedulingverfahren für die ersten 18 Millisekunden an, wann welcher Prozess aktiv ist. 4
- c Erläutern Sie, was man im Zusammenhang mit Schedulingstrategien unter der Antwortzeit eines Prozesses versteht, und bewerten Sie zwei ausgewählte Schedulingstrategien hinsichtlich Fairness und mittlerer Antwortzeit. 2

(Fortsetzung nächste Seite)

5 Das Erzeuger-Verbraucher-Problem ist eine klassische Aufgabenstellung der Informatik.

a Beschreiben Sie, welche Szenarien beim Erzeuger-Verbraucher-Problem durch Synchronisation vermieden werden sollen.

2

Das Erzeuger-Verbraucher-Problem soll mittels folgender Prozesse gelöst werden:

|                                                                                                                                                                                                        |                                                                                                                                                                                                                 |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Erzeuger:</b><br>wiederhole immer<br>erzeuge Element<br>wiederhole<br>lies s<br>warte 10 ms<br>solange s gleich max<br>lege Element im Speicher ab<br>erhöhe s um 1<br>schreibe s<br>endeWiederhole | <b>Verbraucher:</b><br>wiederhole immer<br>wiederhole<br>lies s<br>warte 10 ms<br>solange s gleich 0<br>entnimm Element aus Speicher<br>verringere s um 1<br>schreibe s<br>verbrauche Element<br>endeWiederhole |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

b Geben Sie an, wofür die Variable s in den obigen Prozessen steht, markieren Sie jeweils den Beginn und das Ende des kritischen Abschnitts in den Prozessen und erläutern Sie, warum es bei nebenläufiger Ausführung zu Inkonsistenzen kommen kann.

3

c Um Inkonsistenzen zu vermeiden, soll mit Semaphoren gearbeitet werden. Es wird ein binärer Semaphor für den exklusiven Speicherzugriff eingeführt. Zudem werden zwei Zählsemaphore für die Anzahl an Elementen im Speicher bzw. für die Anzahl an freien Speicherplätzen angelegt.

3

Geben Sie die beschriebenen Semaphore mit ihren Startwerten sowie die entsprechend modifizierten Prozesse für den Erzeuger und den Verbraucher in Pseudocode an.

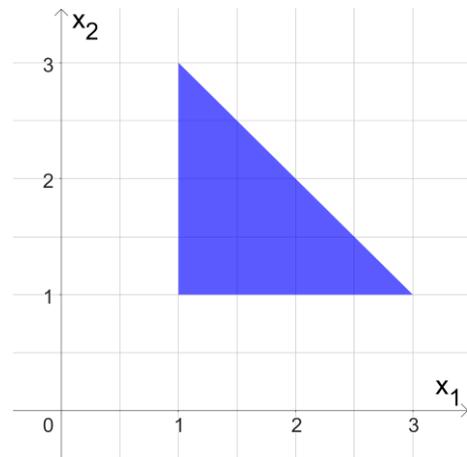
40

### III

BE

- 1 Um die Sicherheit beim Produktionsprozess in einem Chemiewerk zu gewährleisten, dürfen die Wertepaare  $(x_1|x_2)$  zweier Kenngrößen nicht außerhalb des markierten Dreiecks liegen.

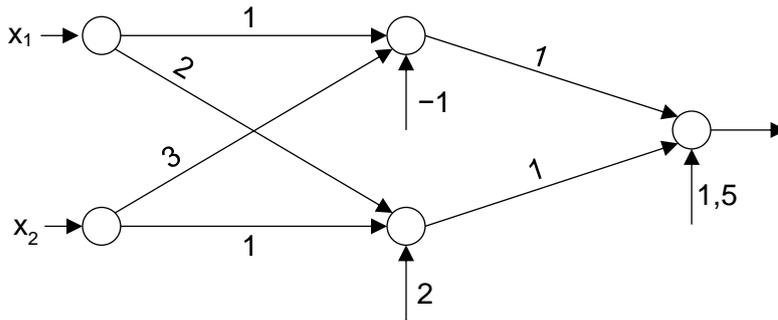
Ein künstliches neuronales Netz soll dazu für Punkte des markierten Dreiecks den Wert 1 und ansonsten den Wert 0 liefern. Als mögliche Aktivierungsfunktionen stehen die Heaviside- und die Identitätsfunktion zur Verfügung.



- a Beschreiben Sie allgemein den Aufbau eines möglichst einfachen, geeigneten künstlichen neuronalen Netzes zur Lösung der beschriebenen Problemstellung. Skizzieren Sie ein solches Netz mit passenden Gewichten. 6
- b Die Software zur Steuerung und Überwachung der Anlage ist über das Firmennetzwerk an das Internet angebunden, sodass Mitarbeitende auch im Homeoffice arbeiten können. Beschreiben Sie eine mögliche Gefährdungssituation durch Personen von außen mit einem Angriffsszenario und benennen Sie zwei Maßnahmen zur Sicherung des Systems. 2
- 2 An einer anderen Stelle im Produktionsprozess werden zwei Flüssigkeiten mit unterschiedlichen pH-Werten miteinander vermischt; je nach Konzentration der beiden Flüssigkeiten ist eine andere Temperatur beim Mischen notwendig. Die Anlage wurde bisher von einem Mitarbeiter überwacht und gesteuert; künftig soll ein künstliches neuronales Netz zur Kontrolle und Regulierung der Parameter über ein IoT-System eingesetzt werden. Die Zusammenhänge sind hier a priori nicht bekannt, allerdings liegt eine große Menge an Lerndaten in Form von Systemparameterwerten mit der jeweiligen Reaktion des Mitarbeiters vor.
- a Beschreiben Sie, wie das Prinzip des überwachten Lernens eingesetzt werden kann, um ein solches künstliches neuronales Netz zu trainieren. 2
- b Geben Sie notwendige Sensoren und Aktoren des IoT-Systems an. 1

(Fortsetzung nächste Seite)

c Die folgende Abbildung zeigt das künstliche neuronale Netz vor der Lernphase.



In der Tabelle sind Eingabewerte und Sollwerte der Ausgabe dargestellt.

| $x_1$ | $x_2$ | Sollwert |
|-------|-------|----------|
| 1     | 0     | 1        |
| 0     | 0     | 0        |
| 0     | 1     | 1        |
| 1     | 1     | 1        |
| 2     | 1     | 0        |

Führen Sie eine Forward Propagation zur Berechnung der tatsächlichen Ausgabewerte des künstlichen neuronalen Netzes durch. Beschreiben Sie, welchen Zweck eine Kostenfunktion im künstlichen neuronalen Netz hat. Geben Sie eine passende Kostenfunktion an und berechnen Sie deren Wert für das Beispiel.

d Beschreiben Sie zwei Möglichkeiten zur Verringerung des Fehlers für den Fall, dass sich nach Anwendung des Backpropagation-Verfahrens keine hinreichende Modellgenauigkeit ergibt.

e Bevor das trainierte künstliche neuronale Netz produktiv eingesetzt wird, soll es zunächst in einer Sandbox installiert werden. Beschreiben Sie, was das für den betroffenen Mitarbeiter bedeutet und wie in dieser Phase zu verfahren ist.

3 Für die Mitarbeitenden im Chemiewerk sollen Arbeitshosen hergestellt werden. Ausgangspunkt ist eine Befragung, bei der die Konfektionsgröße in der Form Weite/Länge (z. B. 32/30; Einheit inch) angegeben wird. Ziel ist es, zur Minimierung der Kosten nur drei unterschiedliche Hosengrößen S, M und L zu produzieren, die den Bedürfnissen der Mitarbeitenden möglichst gut gerecht werden.

Eine Umfrage liefert für die Mitarbeitenden folgende Wertepaare:

| Name  | Curie | Dalton | Nobel | Pauling | Mariotte | Bunsen | Hahn | Haber |
|-------|-------|--------|-------|---------|----------|--------|------|-------|
| Weite | 35    | 35     | 38    | 33      | 40       | 42     | 31   | 28    |
| Länge | 32    | 31     | 35    | 31      | 32       | 32     | 35   | 30    |

a Beschreiben Sie das k-Means-Verfahren und geben Sie an, wie es zur Lösung des gegebenen Problems verwendet werden kann. Geben Sie zudem an, welche Art maschinellen Lernens dem Algorithmus zugrunde liegt, und begründen Sie Ihre Antwort.

b Geben Sie für die erste Iteration und die vorgegebenen Clusterzentren A(35|33), B(40|33) und C(35|35) eine Zuordnung der Mitarbeitenden zu den Clusterzentren an. Bestimmen Sie danach die neuen Clusterzentren.

(Fortsetzung nächste Seite)

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |   |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| <p><b>c</b> Erörtern Sie, welche Chancen und Risiken sich für Unternehmen und Mitarbeitende aus der Erfassung und automatischen Analyse der Konfektionsgrößen ergeben. Gehen Sie auch darauf ein, wie sich die Wahl des Hyperparameters <math>k</math> auswirkt. Nehmen Sie aus der Sicht eines Datenschutzbeauftragten abschließend Stellung zur Datenerfassung.</p>                                                                                                                                                                                                                                                                       | 4 |
| <p><b>4</b> Im Chemiewerk sollen die Beziehungen zwischen Mitarbeitenden und Abteilungen über ein logisches Programm abgebildet werden. Zunächst wird ein Prototyp mit den Personen Curie, Dalton, Nobel, Pauling und Boyle erstellt. Dafür gelten die folgenden Fakten:</p> <ul style="list-style-type: none"> <li>• Curie, Dalton, Nobel bzw. Pauling leiten jeweils die Abteilungen Zentralmanagement, Fertigung, Entwicklung bzw. Buchhaltung.</li> <li>• Boyle ist Mitarbeiter in der Abteilung Fertigung.</li> <li>• Das Zentralmanagement ist den Abteilungen Fertigung, Entwicklung und Buchhaltung direkt übergeordnet.</li> </ul> |   |
| <p><b>a</b> Geben Sie die Wissensbasis mit den beschriebenen Fakten an.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 3 |
| <p><b>b</b> Formulieren Sie die folgenden Aussagen in einer Regel. Beachten Sie, dass es in den Abteilungen des Unternehmens mehrere Hierarchieebenen geben kann:</p> <ul style="list-style-type: none"> <li>• Eine Abteilung ist auch über mehrere Ebenen hinweg einer anderen Abteilung übergeordnet.</li> <li>• Die Leitung einer Abteilung ist Vorgesetzte für alle Mitarbeitenden der Abteilung sowie für die Leitung und Mitarbeiter untergeordneter Abteilungen.</li> </ul>                                                                                                                                                          | 6 |
| <p><b>c</b> Entwerfen Sie eine Anfrage an die Wissensbasis, um zu ermitteln, für welche Mitarbeitenden Curie Vorgesetzte ist.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 1 |

1 Die internationale Standardbuchnummer, kurz ISBN, dient der eindeutigen Kennzeichnung von Büchern. Einer ihrer Bestandteile ist die ein- bis fünfstellige Gruppennummer, die eine Kennzahl für den Sprachbereich ist. Die zulässigen Bereiche, in denen die Gruppennummer liegen kann, sind in folgender Übersicht dargestellt:

- einstellige Gruppennummern: 0 – 5 und 7
- zweistellige Gruppennummern: 80 – 94
- dreistellige Gruppennummern: 600 – 649 und 950 – 989
- vierstellige Gruppennummern: 9900 – 9989
- fünfstellige Gruppennummern: 99900 – 99999

Die Menge aller zulässigen Gruppennummern wird als Sprache  $L$  über dem Alphabet  $\Sigma = \{0; 1; \dots; 9\}$  aufgefasst.

a Geben Sie das Zustandsübergangdiagramm eines endlichen, erkennenden Automaten an, der genau die Gruppennummern der Sprache  $L$  akzeptiert. 5

b Geben Sie die Produktionsregeln einer formalen Grammatik an, mit denen  $L$  erzeugt werden kann. 3

c Es sollen nun ohne Einschränkung Nummern beliebiger Länge aus den Ziffern 0 bis 9 betrachtet werden. Nach jeder Nummer sollen so viele Zeichen  $_$  angefügt werden, wie die Nummer Ziffern hat. 9

Die Menge der so gebildeten Wörter bildet die formale Sprache  $L^*$ . Zu  $L^*$  gehören z. B.  $3_$  und  $028_ _ _$ .

Geben Sie eine Turingmaschine an, die genau dann terminiert, wenn ein Wort der Sprache  $L^*$  auf dem Band liegt.

2 Zur Berechnung von Zweierpotenzen mit nichtnegativen, ganzzahligen Exponenten wird der folgende rekursive Algorithmus betrachtet:

```

Methode zweiHoch(n)
 wenn n gleich 0
 gib 1 zurück
 sonst
 wenn n gerade ist
 gib zweiHoch(n/2) * zweiHoch(n/2) zurück
 sonst
 gib 2 * zweiHoch((n-1)/2) * zweiHoch((n-1)/2) zurück
 endeWenn
 endeWenn
endeMethode

```

*Hinweis:* Der Operator „/“ liefert den Wert des ganzzahligen Quotienten (ohne Rest). Beispielsweise gilt:  $7 / 2 = 3$ .

a Zeigen Sie, dass der Algorithmus für  $n = 6$  das richtige Ergebnis liefert, indem Sie *zweiHoch(6)* als Folge von Methodenaufrufen angeben. 2

(Fortsetzung nächste Seite)

- b** Geben Sie an, wie oft die Methode *zweiHoch* bei Eingabe der Werte 0, 1, 3, 6 und 12 jeweils insgesamt aufgerufen wird. 4
- Bestimmen und erklären Sie, wie sich die Anzahl der Methodenaufrufe allgemein bei Verdopplung des Eingabewerts ändert. Geben Sie außerdem eine Vermutung für den Laufzeitaufwand von *zweiHoch* in Abhängigkeit von  $n$  unter Verwendung der Landau-Notation an.
- c** Zweierpotenzen mit nichtnegativen, ganzzahligen Exponenten können auch mithilfe der folgenden Methode *zweiHochB* berechnet werden. 3
- ```

Methode zweiHochB(n)
  wenn n gleich 0 dann
    gib 1 zurück
  sonst
    gib 2 * zweiHochB(n-1) zurück
  endeWenn
endeMethode

```
- Geben Sie den Laufzeitaufwand des in der Methode *zweiHochB* gegebenen Algorithmus in Abhängigkeit von n an. Verwenden Sie dazu die Landau-Notation.
- Entscheiden Sie begründet, welcher der beiden Methoden *zweiHoch* oder *zweiHochB* Sie zur Berechnung von Zweierpotenzen den Vorzug geben würden.
- 3** Eine Version des Rucksackproblems kann wie folgt formuliert werden.
- Gegeben:*
Eine positive natürliche Zahl b und dazu n positive natürliche Zahlen $a_1, a_2, a_3, \dots, a_n$.
- Problemstellung:*
Ist es möglich, aus den n positiven natürlichen Zahlen $a_1, a_2, a_3, \dots, a_n$ einige so auszuwählen, dass deren Summe b ergibt? Kann also eine Teilmenge $I \subset \{1, 2, \dots, n\}$ so gewählt werden, dass $\sum_{i \in I} a_i = b$?
- a** Für $n=6$ sind $a_1=13, a_2=13, a_3=13, a_4=14, a_5=14$ und $a_6=14$ gegeben. 2
- Zeigen Sie, dass es für $b=40$ eine geeignete Auswahl an Werten gibt. Zeigen Sie außerdem, dass das für $b=43$ nicht der Fall ist.
- b** Formulieren Sie einen Brute-Force-Algorithmus in Pseudocode, der prüft, ob in obigem Rucksackproblem die Bildung des Summenwerts b mit einer geeigneten Auswahl der n positiven ganzen Zahlen möglich ist. 8
- c** Nun wird folgendes Problem betrachtet: 2
- Es soll entschieden werden, ob mit den Zahlen 13, 14 und 17 die Summe b gebildet werden kann, wobei jede Zahl beliebig oft als Summand gewählt werden darf.
- Führen Sie dieses Problem auf das oben beschriebene Rucksackproblem zurück.
- d** Das Rucksackproblem gehört zur Komplexitätsklasse NP. Grenzen Sie die beiden Komplexitätsklassen P und NP voneinander ab. 2