



(a)

Zähle alle Wörter mit maximaler Länge 3 auf, die aus den Symbolen des Alphabets  $\Sigma = \{A, B\}$  gebildet werden können.

(b)

Zeichne einen DFA über dem Alphabet  $\Sigma = \{0, 1\}$ , der alle Wörter  $w$  mit  $|w| \geq 2$  akzeptiert, die mit einer Null beginnen und in denen kein Zeichen zweimal nacheinander vorkommt.



- (a) Bestimme die Anzahl der Vergleiche, die der „naive“ Algorithmus für das String-Matching zum Auffinden des Musters **GGCA** im Textstring **GGGAAAGGCAT** benötigt.

G	G	G	A	A	A	G	G	C	A	T					

- (b) Leite die Worst-Case-Laufzeit für die Suche nach einem Muster  $p$  der Länge  $m$  in einem Text  $t$  der Länge  $n$  anhand eines geeigneten Beispiels her.

## Übungsfragen 3

## Thema A

(a)

Vervollständige die Methoden der Python-Klasse Stack.

```
class Stack:
```

```
    def __init__(self):  
        self.items = []
```

```
    def push(self, item):  
        ...
```

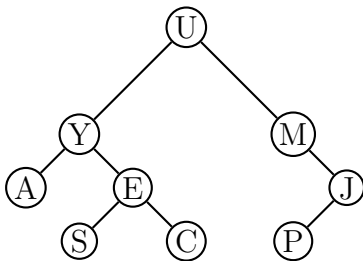
```
    def pop(self):  
        ...
```

```
    def is_empty(self):  
        ...
```

```
    def size(self):  
        ...
```

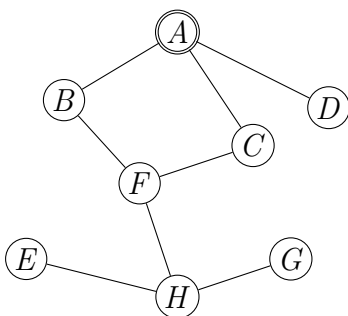
(b)

Gib die Folge der Knoten an, wenn der folgende Baum in (a) Preorder-Reihenfolge (b) Postorder-Reihenfolge durchlaufen wird.



(c)

Führe eine Breitensuche auf dem gegebenen Graphen durch, wobei Nachbarknoten in alphabetischer Reihenfolge besucht werden sollen.



## Übungsfragen 3

## Thema B

(a)

Komprimiere die Zeichenfolge

AAAAAAABB!CCCCCCCCCCCCCCCC

mit der Lauflängencodierung. Verwende '!' als Steuerzeichen und eine Ziffer als Zählzeichen.  
Wie gross ist die Kompressionsrate?

(b)

Komprimiere die Zeichenfolge AAAAAAAAAABC mit dem LZW-Verfahren und bestimme die Kompressionsrate.

Gegeben ist das folgende Python-Modul.

```
class Image:

    def __init__(self, width, height):
        self.w = width
        self.h = height
        self.img = [[0 for i in range(width)] for j in range(height)]

    def set_pixel(self, x, y, color=1):
        self.img[y][x] = color

    def write(self, filename):
        fd = open(filename, mode='w')
        fd.write('P1\n')
        fd.write('{0} {1}\n'.format(self.w, self.h))
        for i in range(0, self.h):
            for j in range(self.w):
                fd.write('{0} '.format(self.img[i][j]))
        fd.close()

I = Image(3, 3)
I.set_pixel(0, 0)
I.set_pixel(1, 1)
I.set_pixel(2, 2)
I.write('test.pbm')
```

- (a) Beschreibe, was dieses Modul macht.
- (b) Gibt es Daten aus? Wenn ja, welche Daten und in welcher Form?





(a)

Skizziere schematisch die Situation, in der sich zwei Kommunikationspartner befinden, die eine vertrauliche Informationen austauschen wollen und beschreibe sie mit Fachbegriffen.

(b)

Alice und Bob einigen sich auf die Primzahl  $p = 17$  und den Generator  $g = 10$ . Alice wählt die Zahl  $a = 2$  und Bob die Zahl  $b = 16$ . Wie lautet der gemeinsame Schlüssel aufgrund des Diffie-Hellman-Schlüsselautauschprotokolls?

Verwende den Taschenrechner so weit es geht.

(a)

Erstelle auf der Grundlage der folgenden Angaben das ER-Modell einer Firma, die Informatikprojekte durchführt. Bestimme selber geeignete Attribute und notiere die richtigen Beziehungstypen (Kardinalitäten).

- Die Firma besteht aus mehreren Abteilungen und hat mehrere Mitarbeiter. Jeder Mitarbeiter gehört zu genau einer Abteilung.
- An jedem Projekt arbeitet mindestens ein Mitarbeiter. Es gibt auch Mitarbeiter, die nicht an einem Projekt arbeiten.

(b)

Welche Resultattabellen erzeugen die SQLite-Anweisungen bei folgender Tabelle?

tab

<u>id</u>	a	b	c
1	1	5.3	t
3	0	7.6	s
4	4	3.2	t
5	1	4.1	s
6	4	6.5	r

(a) `SELECT AVG(a) FROM tab;`

(b) `SELECT b, c FROM tab WHERE id > 3 ORDER BY b;`

(c) `SELECT SUM(a), c FROM tab GROUP BY c;`

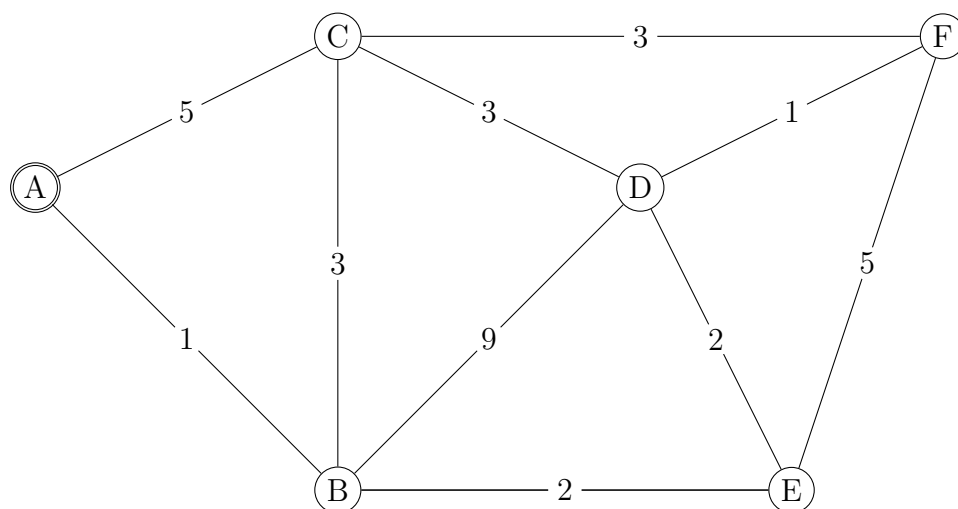
## Übungsfragen 6

## Thema B

(a)

Bestimme, ausgehend vom Knoten  $A$ , die kürzesten Wege zu den übrigen Knoten mit dem Algorithmus von Dijkstra und kennzeichne dann den Spannb Baum.

*Wichtig:* Für einen nachvollziehbaren Lösungsweg müssen auch die gestrichenen (relaxierten) Zwischenresultate sichtbar sein. Falls bei einem Schritt mehrere Knoten als Arbeitsknoten in Frage kommen, kann ein beliebiger ausgewählt werden.



(b)

Nenne zwei Anwendungen für den Algorithmus von Dijkstra