

# ABSCHLUSSKLAUSUR ZU PROGRAMMIERUNG 1 IM WINTERSEMESTER 2012/13



Geschrieben am 08.02.2013

Vorname \_\_\_\_\_  
Nachname \_\_\_\_\_  
Matrikelnummer \_\_\_\_\_  
Geburtsdatum \_\_\_\_\_  
Studiengang \_\_\_\_\_

Klausur für 9 Credit Points werten? Ja ☐

Vom Prüfer auszufüllen:

Aufgabe	1	2	3	4	5	6	7	8	9	10	$\Sigma$
Punkte	10	10	10	10	10	10	10	10	10	10	100
Davon erreicht											

Klausurpunkte \_\_\_\_\_  
+ Bonuspunkte \_\_\_\_\_  
 $\Sigma$  \_\_\_\_\_  
**Note** \_\_\_\_\_

## Generelle Klausurhinweise:

1. Geben Sie auf jedem Blatt (oben rechts) Ihre Matrikelnummer an. Blätter ohne Matrikelnummer können nicht gewertet werden.
2. Schreiben Sie bitte *leserlich*!
3. Kontrollieren Sie Ihre Klausur auf Vollständigkeit. Die Seitenzahlen befinden sich unten rechts.
4. Verwenden Sie die Rückseiten der Klausur ausschließlich für eigene Notizen - diese werden **nicht** gewertet. Die letzte Seite der Klausur ist als "*Schmierpapier*" vorgesehen, oder falls der Platz zum Beantworten einer Frage nicht ausreicht. Zur Benotung muss ein Verweis bei der Aufgabenstellung und eine deutliche Kennzeichnung auf dem Schmierblatt enthalten sein. Benötigen Sie weiteres Papier, melden Sie sich bei der Aufsicht. Selbst mit gebrachtes Papier wird als Täuschungsversuch gewertet!
5. Außer einem dokumentenechten Stift - kein Bleistift - (**nicht** Rot) sind keine weiteren Hilfsmittel zugelassen, wie Handy, Smartphone, Taschenrechner, Laptop etc. Ein betriebsbereites Handy oder Smartphone wird als Täuschungsversuch gewertet.
6. Die Prüflinge können während der Klausur einzeln die Toilette besuchen. Vor Verlassen des Klausorraumes haben diese bei der Aufsicht ihren Namen anzugeben.
7. Für die Bearbeitung der Klausur stehen 180 Minuten zur Verfügung. In der letzten halben Stunde (30 Minuten) vor Abgabe ist es den Prüflingen untersagt den Raum zu verlassen, um unnötige Unruhe zu vermeiden.

**Aufgabe 1: Multiple Choice****Punkte: 10/ \_\_**

Eine Frage gilt dann als korrekt beantwortet, wenn alle richtigen und keine falschen Antworten angekreuzt sind. Mindestens eine Antwort ist immer richtig.

**(a) (1 Punkt) Algorithmus**

Algorithmen, die zu jedem Zeitpunkt der Ausführung maximal eine Möglichkeit der Programmfortsetzung besitzen, sind:

- ☐ Determiniert
- ☐ Determinante
- ☐ Determination
- ☐ Deterministisch

**(b) (1 Punkt) Formale Sprachen**

Gegeben sei die folgende Grammatik in EBNF mit Startsymbol S.

```
S = {P} | {M};  
P = P, "+", {M} | Z, {Z};  
M = M, "*", M | Z, {Z};  
Z = "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9";
```

Welche Wörter gehören zu der durch die Grammatik definierten Sprache?

- ☐ 10\*20
- ☐ 64+32+
- ☐ P+MMM
- ☐ 128+64\*32

**(c) (1 Punkt) Rechner-Architektur**

Welche der folgende Komponenten gehören zur von-Neumann-Architektur?

- ☐ Lesewerk
- ☐ Hauptwerk
- ☐ Rechenwerk
- ☐ Speicherwerk

**(d) (1 Punkt) Pythonanalyse**

Was berechnet der folgende Python-Code?

```
1 x, y = 7, 21  
2 x += x // y * 3.1  
3 print(x)
```

- ☐ 7
- ☐ 8
- ☐ 7.0
- ☐ NameError: name is not defined

(e) (1 Punkt) **Python Ausdrücke**

Welche der folgenden Ausdrücke werden von Python (Version 3.x) zu **True** ausgewertet?

- ☐ `bool(12 % 5.0)`
- ☐ `bool(1 >> 2 >> 3)`
- ☐ `bool(1.0 == 0 and 0.0 != 1.0)`
- ☐ `bool(not True or 12 * 12 / 12)`

(f) (1 Punkt) **Built-in Datentypen**

Welche der folgenden Python-Builtins sind veränderlich (*mutable*)?

- ☐ `Set`
- ☐ `Tuple`
- ☐ `String`
- ☐ `Dictionaries`

(g) (1 Punkt) **Listen**

Was gilt für Pythons Listen?

- ☐ Listen sind immer sortiert
- ☐ Listen dürfen nicht leer sein
- ☐ Listenelemente sind immer eindeutig
- ☐ Listen können beliebige Elemente verwalten

(h) (1 Punkt) **Funktionen & Prozeduren**

Welche Unterschiede gelten für Funktionen und Prozeduren in Python?

- ☐ Python trifft eine statische Unterscheidung
- ☐ Python ergänzt Prozeduren automatisch zu Funktionen
- ☐ Python macht eine Pseudounterscheidung mit Hilfe von `--`
- ☐ Python verwendet die Argumentenliste zur Unterscheidung

(i) (1 Punkt) **Objektorientierung**

Welche Relation besteht zwischen einer Klasse und ihrer Basisklasse?

- ☐ Komposition
- ☐ Spezifikation
- ☐ Lokalisierung
- ☐ Generalisierung

(j) (1 Punkt) **Prozess**

Woraus besteht ein *Prozess* unter anderem?

- ☐ Stack
- ☐ Scheduler
- ☐ Prozesskontext
- ☐ Programmdateien

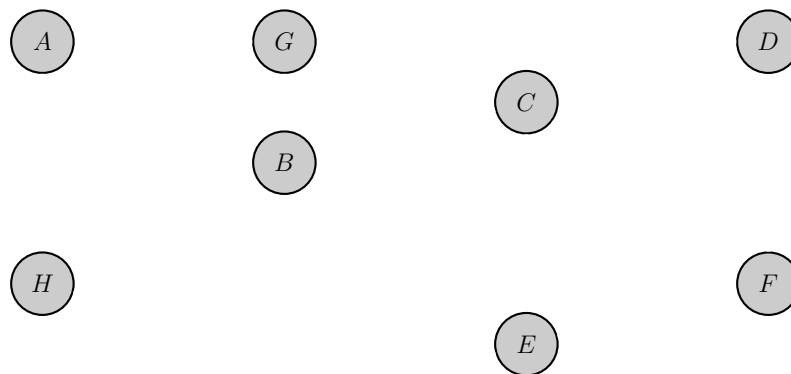




**Aufgabe 3: Python-Datentypen****Punkte: 10/ \_\_**

- (a) (1 Punkt) Rekonstruieren Sie den gerichteten Graphen  $\mathcal{G}$  anhand der Adjazenzliste  $L$  für die Knoten A, B, C, D, E, F, G und H.

```
L = [ [A, [H, D]],
       [B, []],
       [C, [B, E, F]],
       [D, [G]],
       [E, [H, F]],
       [F, [D]],
       [G, [A, B]],
       [H, [C]] ]
```



- (b) (1 Punkt) Geben Sie die Adjazenzmatrix  $\mathcal{M}$  des Graphen  $\mathcal{G}$  an<sup>1</sup>.

$$\mathcal{M} = \begin{matrix} & \begin{matrix} A & B & C & D & E & F & G & H \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{matrix} & \left( \begin{array}{cccccccc} & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \end{array} \right) \end{matrix}$$

---

<sup>1</sup>Die Matrix muss nicht mit Nullen aufgefüllt werden.

(c) (2 Punkte) Gegeben sind die folgenden Python Codezeilen:

```
1 a = 2 * 3 ** 80 + 1
2 b = 2 << 42 // 15 % 3
3 c = 4 / (4 - 2 * 3)
4 d = 1 < 2 % 3 + 4
```

Tragen Sie die Operatoren in der Reihenfolge in die Kästchen ein, in der Sie ausgewertet werden.

Der Operator wird als 1. 2. 3. ausgewertet.

Zeile 1	<input type="text"/>	<input type="text"/>	<input type="text"/>
Zeile 2	<input type="text"/>	<input type="text"/>	<input type="text"/>
Zeile 3	<input type="text"/>	<input type="text"/>	<input type="text"/>
Zeile 4	<input type="text"/>	<input type="text"/>	<input type="text"/>

(d) (1 Punkt) Von welchem **type** sind die Variablen **c** und **d**?

1. c

1. \_\_\_\_\_

2. d

2. \_\_\_\_\_

(e) (1 Punkt) Was sind die Operatoren für die folgenden Operationen in Python?

1. Bitweise Konjunktion

1. \_\_\_\_\_

2. Logische Disjunktion

2. \_\_\_\_\_

(f) (1 Punkt) Wie lauten die größte und die kleinste Zahl im Zweierkomplement, die sich mit einer Blockgröße von n-Bit darstellen lassen?

(f) \_\_\_\_\_

(g) (1 Punkt) Welchen Text gibt der nachstehende **print**-Befehl aus?

```
print("irQeelGaftPetspäfKw NedsDrAiVH"[-1:0:-2])
```

(g) \_\_\_\_\_

(h) (1 Punkt) Wie wird die implizite Typkonvertierung bezeichnet?

(h) \_\_\_\_\_

(i) (1 Punkt) Was ist der Vorteil der erweiterten Zuweisungen?

\_\_\_\_\_  
\_\_\_\_\_







**Aufgabe 5: OOP & OOAD****Punkte: 10/ \_\_\_\_**

(a) (3 Punkte) Zeichnen Sie die UML-Symbole für die folgenden Elemente:

**1. Assoziation**


**2. Aggregation**


**3. Komposition**


**4. Generalisierung**


**5. Synchroner Operationsaufruf**


**6. Antwort des Synchronen Operationsaufrufes**


(b) (1 Punkt) Mit welchen *besonderen* Methoden werden Objekte *im Allgemeinen* instantiiert, bzw. wird der von einem Objekt belegte Speicher freigegeben?

---



---



---



---

(c) (2 Punkte) Erläutern Sie den Begriff „*Überschreiben*“ im Kontext der objekt-orientierten Programmierung.

---



---



---



---



---



---



---



---

- (d) (2 Punkte) Was ist notwendig, um von *Polymorphie* im Kontext der objektorientierten Programmierung zu sprechen?

---

---

---

---

---

---

- (e) (1 Punkt) Wie viele Referenzen kann es in Python auf ein Objekt geben?

---

---

---

---

---

---

- (f) (1 Punkt) Welche Sichtbarkeitsmodifikatoren gibt es in der UML und mit welchen Symbolen werden sie repräsentiert?

---

---

---

---

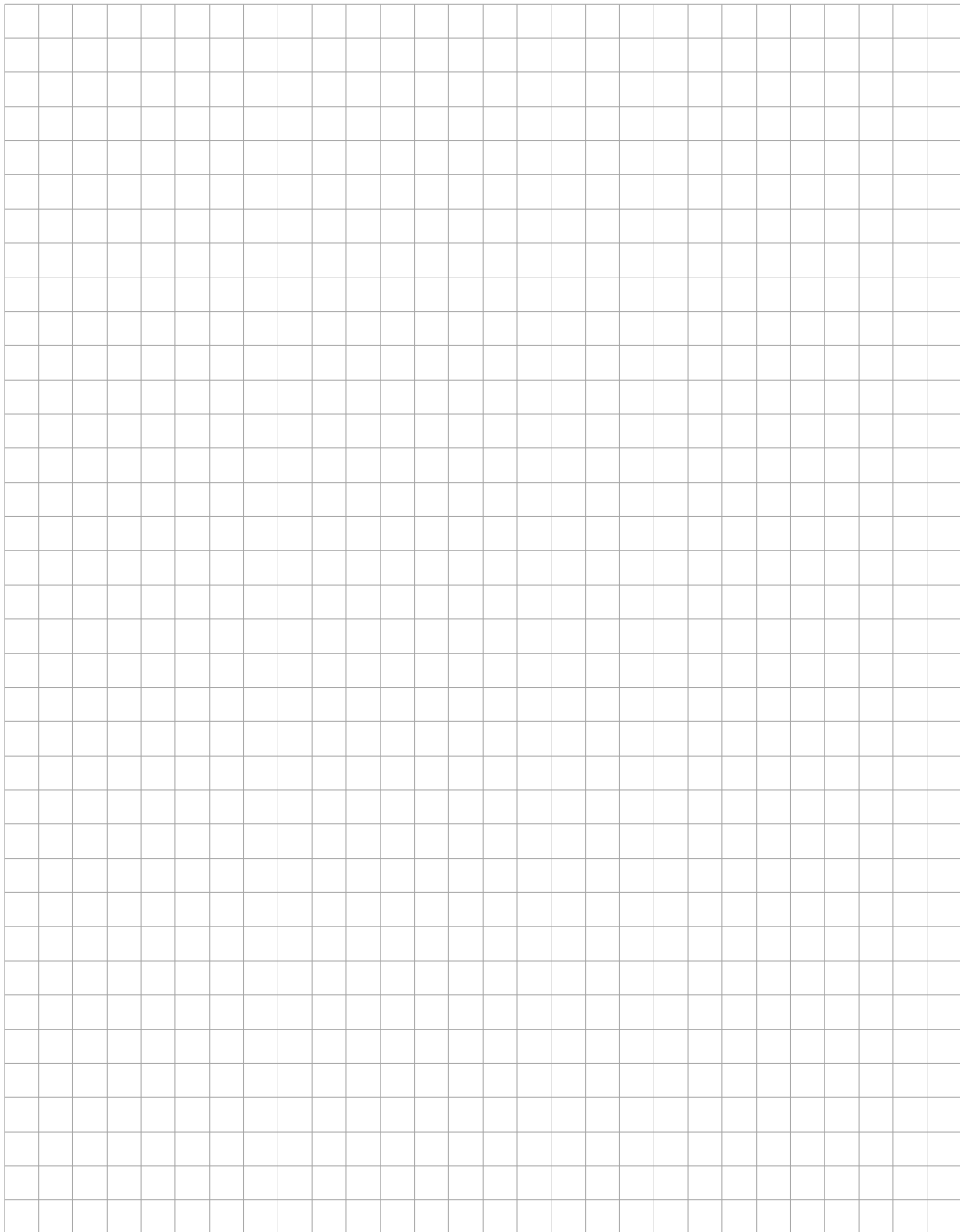
---

---

**Aufgabe 6: Testen****Punkte: 10/** \_\_\_\_\_

(a) (1 Punkt) Übertragen Sie den nachstehenden Code in ein Flussdiagramm.

```
1 a, b, c = input("a, b, c")
2 while True:
3     if a < b:          ## B1
4         a += 4        ## A1
5     else:
6         if a % 2:      ## B2
7             a += 1     ## A2
8         else:
9             a -= 10    ## A3
10    if a <= c:          ## B3
11        break          ## A4
```



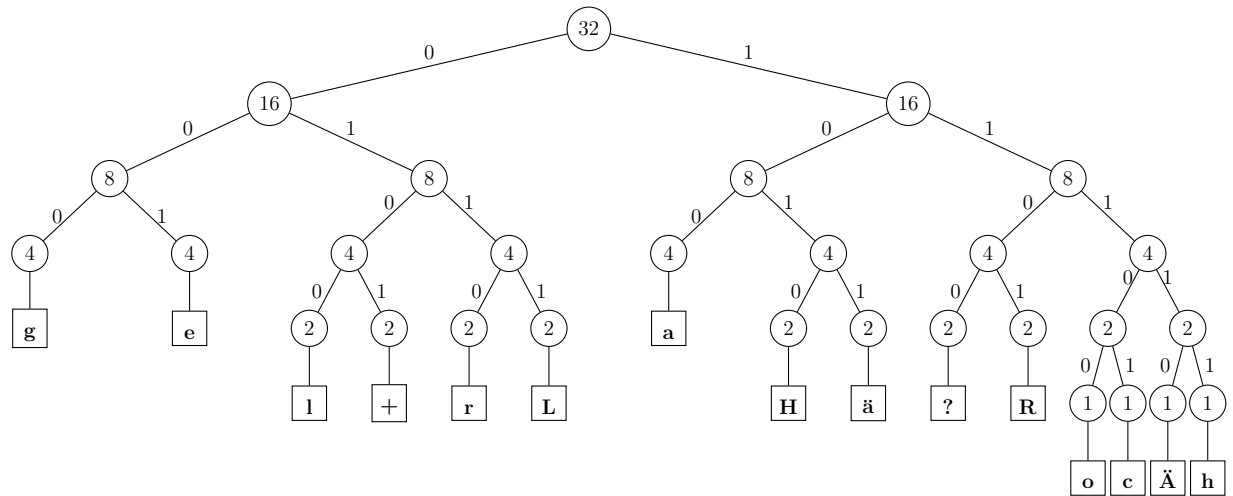
- (b) (2 Punkte) Geben Sie Testfälle für eine Zweigüberdeckung des Codes aus Aufgabenteil [a](#) an. Benennen Sie die Reihenfolge, in der die Anweisungen und Bedingungen (B1, B2, B3) durchlaufen werden.

This image shows a full page of blank graph paper. The grid consists of small, uniform squares formed by thin, light gray lines. There are no margins, text, or other markings on the page.

- (c) (1 Punkt) Geben Sie eine formalisierte Testmenge für eine Anweisungsüberdeckung aus  $a$  an.

This image shows a full page of blank graph paper. The grid consists of small, uniform squares formed by thin, light gray lines. There are no margins, text, or other markings on the page.



**Aufgabe 7: Daten – Information – Wissen****Punkte: 10/** \_\_\_\_\_Gegeben sei der folgende Code-Baum einer *Huffman-Kodierung*.(a) (2 Punkte) Dekodieren Sie die Nachricht<sup>2</sup>.

1010111001 1101111110 1001000000 0101100101 1101001000 1000111010  
 1011110000 0001011011 0100100010 0010010101 0111011111 1011001100

\_\_\_\_\_

\_\_\_\_\_

(b) (1 Punkt) Vervollständigen Sie die Tabelle.

Zeichen $z$	Anzahl	$H_n(z)$	$I(z)$
a			
c			
e			
ä			
g			
H			
+			
l			
o			
L			
r			
h			
R			
?			
Ä			

<sup>2</sup>Die in der Nachricht enthaltenen Leerzeichen dienen der Lesbarkeit und sind nicht Bestandteil der Nachricht.







- (d) (1 Punkt) Atomare Aktionen lassen sich mittels Hardware implementieren. Welche Verfahren wurden hierzu in der Vorlesung diskutiert?

---

---

---

---

---

- (e) (1 Punkt) Wodurch wird das Kontrollflussprinzip in imperativen Programmiersprachen implementiert? Nennen Sie Beispiele.

---

---

---

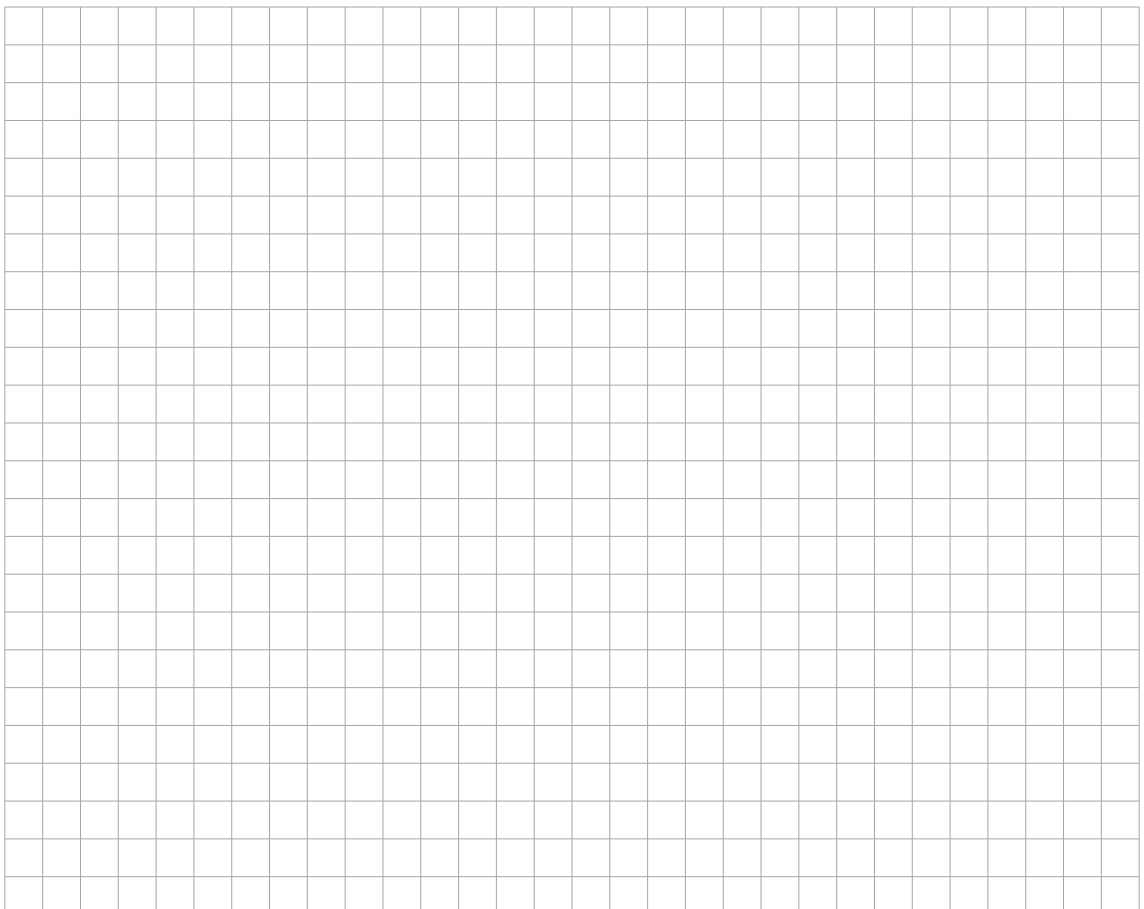
---

---

- (f) (3 Punkte) Zeichnen Sie den Datenflussgraphen für die Funktion

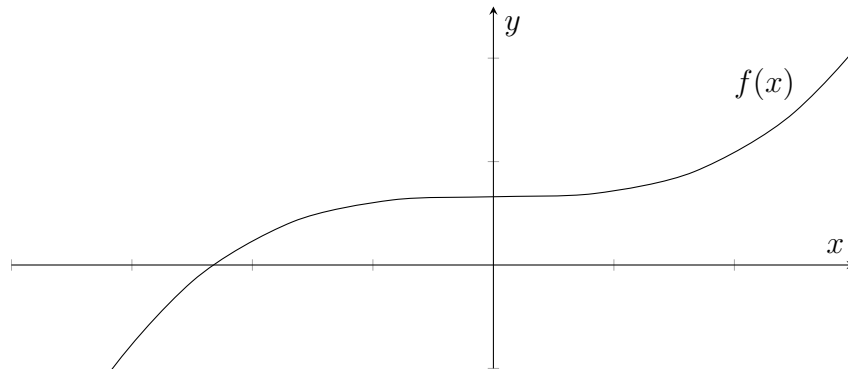
$$5 \cdot x^3 + 7 \cdot y^2 - (4 \cdot x^2 + 2 \cdot x)$$

und geben Sie an, welche der Schritte sich parallel berechnen lassen.

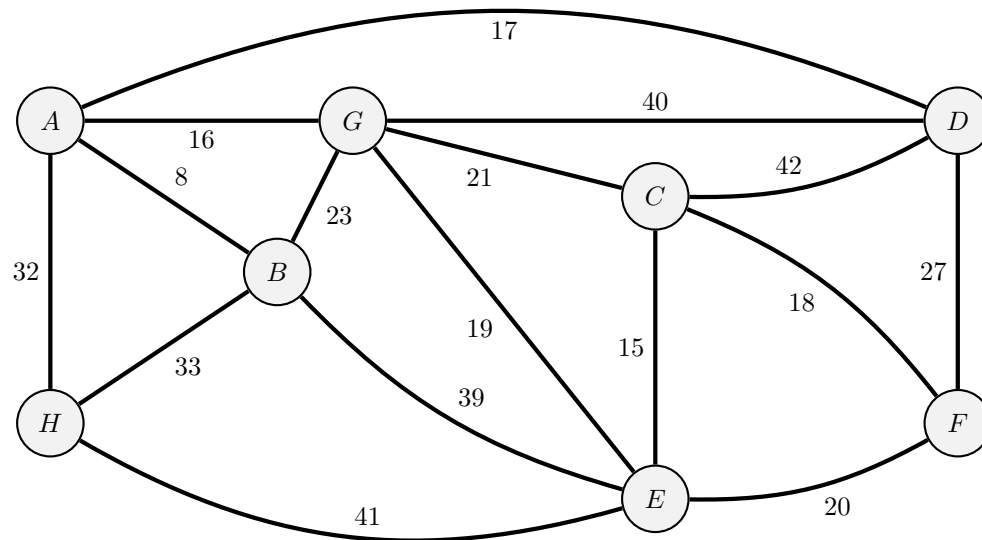


**Aufgabe 9: Algorithmenentwurf****Punkte: 10/ \_\_**

- (a) (5 Punkte) Formulieren Sie einen *Divide & Conquer*-Algorithmus als Anweisungsvorschrift (Pseudocode) zur Ermittlung der Nullstelle einer streng monoton Funktion  $f$  im Intervall  $[a, b]$ .



(b) Gegeben sei der folgende Graph  $\mathcal{G}$ .



1. (2 Punkte) Geben Sie die aufsteigend sortierte Kantenliste an.

Kante								
Gewicht								
Kante								
Gewicht								

2. (3 Punkte) Ermitteln Sie den minimalen Spannbaum mit dem Algorithmus von *Kruskal*. Geben Sie für jeden Rechenschritt die aktuell besuchte Kante, sowie die Knoten der bereits ermittelten Teilgraphen an.

Kante	Knoten der Teilgraphen

Falls Sie auf der Titelseite das Kreuz für 9 CP's gesetzt haben entfällt diese Aufgabe für Sie.

**Aufgabe 10: Softwareentwurf**

**Punkte: 10/ \_\_**

(a) (6 Punkte) Nennen und erklären Sie die einzelnen Phasen des Wasserfallmodells.

**1. Phase**

---

---

---

---

---

**2. Phase**

---

---

---

---

---

**3. Phase**

---

---

---

---

---

**4. Phase**

---

---

---

---

---

**5. Phase**

---

---

---

---

---

**6. Phase**

---

---

---

---

- (b) (1 Punkt) Worin besteht der Sinn der Modularisierung von Softwaresystemen?  
Was bildet die Basis für diese Modularisierung?

---

---

---

---

---

- (c) (2 Punkte) Mit welchen Kriterien lässt sich die Qualität eines modularen Entwurfs „messen“? Was bezeichnen diese?

---

---

---

---

---

- (d) (1 Punkt) Wodurch wird die Adaptierbarkeit eines Moduls begünstigt?

---

---

---

---

---

