GdP-Klausur Feb. 2023

Aufgabe 1 (Grundlagen, Graphen)

1. Es soll ein Straßennetz mittels gerichtetem Graphen modelliert werden. Beschreiben sie an diesem Beispiel die folgenden Eigenschaften der Modellbildung in wenigen Stichpunkten:
   (a) Beschränkung auf wesentliche Aspekte der Realität
   (b) Abstraktion
2. Gegeben sei ein gerichteter Graph G = (V, E) durch die Knotenmenge V = {a, b, c, d, e} und die Kantenmenge E = {(a, b), (a, c), (b, c), (c, d), (c, e), (e, c), (d, b)}.
   (a) Geben sie die Adjazenzmatrix an, die sich aus dem gegebenen Graphen ergibt. Beachten sie dabei die alphabetische Reihenfolge der Matrixeinträge!
   (b) Zeichnen sie den Graphen, der sich ergibt.
   (c) Geben sie alle Kreise, die G enthält.
   (d) Geben sie einen Zykel von G an, der kein Kreis ist.
   (e) Geben sie die Reihenfolge der Knoten an, die abgesucht werden, falls der Graph von Knoten a aus mithilfe der Breitensuche durchlaufen wird.
   (f) -“- mithilfe der Tiefensuche.
3. Geben sie die formale Definition des „kleinsten Abstand zweier Knoten“ an. Sie können dabei die Begriffe „Pfad“ und „Pfadlänge“ ohne weitere Erklärungen verwenden.

Aufgabe 2 (Entwurf von Algorithmen)

Der geforderte Algorithmus soll in Pseudocode aufgeschrieben werden. Dieser muss soweit verfeinert werden, dass eine Übertragung in eine imperative Programmiersprache ohne weitere kreative Schritte möglich ist.

1. Gegeben sei die Adjazenzmatrix eines gerichteten Graphen G = (V, E). Entwerfen sie einen Algorithmus, der die Anzahl der Kanten bestimmt.
2. Der Algorithmus bekommt eine Liste L als Eingabe, deren Elemente positive ganze Zahlen sind. Geprüft werden soll, ob jedes Element in L paarweise verschieden von jedem anderen Element ist. Falls dies zutrifft, soll 1 zurückgegeben werden, sonst 0.

Aufgabe 3 (Ressourcen)

1. Bestimmen sie die Zeitkomplexität des gegebenen Programms im worst case. Dabei sei n die Programmvariable, die durch die Nutzereingabe festgelegt wird.

```
n = int(input("Geben sie eine ganze positive Zahl an: "))
x = 0
z = 0

while (x < y):
    y = n
    while y > 0:
        z += 1
        y = y - 1
    x += 1
```

2. Beweisen sie den folgenden Zusammenhang:

   $2n^4 + 9n^2 - 11 \in \Theta(n^4)$

3. Zwei Algorithmen A und B lösen das gleiche Problem. Die Zeitkomplexitäten von A und B sind t_A und t_B. Es sei $t_A \in O(n^2)$ und $t_B \in \Omega(n^3)$.

   (a) Kreuzen sie alle zutreffenden Aussagen an:

   ☐ $t_A \in O(t_B)$

   ☐ $t_B \in O(t_A)$

   (b) Welcher der Algorithmen ist der effizientere (hinsichtlich der Zeitkomplexität)?

Aufgabe 4 (Imperative & Prozedurale Programmierung)

Gegeben sei das folgende Code:

```
def f(x):
    y = x + 1
    y = g(y)
    return y


def g(x):
    y = 3 ** x
    return y // 4


y = 2
x = f(y)
print(x)
```

1. Geben sie alle formale und aktuelle Parameter an.
2. Welche Ausgabe erfolgt durch die print-Anweisung?
3. Zeichnen sie den Aufrufstack mit allen Stackframes, die beim Aufruf von g entstanden sind. Geben sie dabei auch zu den jeweiligen Zeitpunkten die Werte der Variablen an.

Aufgabe 5 (Objektorientierte Programmierung)

Es sei die folgende Klassendefinition der Klasse „Hund“ gegeben:

```
class Hund:

        def __init__(self, hungrig):
                self.hungrig = hungrig

        def fressen(self, menge):
                self.hungrig = self.hungrig - menge

        def bellen(self):
```

1. Ergänzen sie die Methode „bellen“ sodass der Hund bei Aufruf dieser Methode genau so oft bellt, wie er hungrig ist. Verwenden sie dazu die print()-Funktion
2. Nennen sie alle Datenelemente der Klasse Hund.
3. Es wird angenommen, dass der Hund mit einer Hungerstärke von 10 geboren wird. Schreiben sie ein Codesegment außerhalb der Klassendefinition, sodass der Hund nach der Ausführung des Programms eine Hungerstärke von 6 hat.

Aufgabe 6 (Funktionale Programmierung)

1. Gegeben sei das folgende Codesegment:

   ```
   x = abs(x)
   for i in range(1, x):
       x = x + 4
   print(bin(x))
   ```

   Entspricht dies den Kriterien der funktionalen Programmierung? Wenn nicht, beschriften sie die entsprechenden Codezeilen und erläutern sie dies in 1-2 Stichpunkten.

2. Implementieren sie die das folgende Codesegment rekursiv!

   ```
   def func(n):
       x = 1
       for i in range(1, n+1):
           x = 3 * x
       return(x)
   ```

   Bestimmen sie die Laufzeit ihres Programms möglichst genau! Gehen sie dabei von einem worst case aus.

3. Gegeben sei die Liste L = [2, 4, 1, 6, 7]. Schreiben sie eine höherwertige Funktionen (map und filter), die L als Eingabe bekommen und die folgenden Listen ausgeben:

   (a) [3, 5, 2, 7, 8]

   (b) [2, 4, 6]

4. Schreiben sie eine Python-Funktion „occurs“, die rekursiv eine Sequenz durchsucht. Dieses soll folgende Anforderungen erfüllen:

   Eingabe: Linkssequenz ganzer Zahlen, ganze Zahl x

   Ausgabe: True, falls die Sequenz x enthält, sonst False

Aufgabe 7 (Compiler, Interpreter, AASS)

1. Nennen sie die grundlegende Gemeinsamkeit von Compilern und Interpretern.
2. Nennen sie zwei Unterschiede zwischen Compilern und Interpretern.
3. Wofür wird ein Assembler verwendet?
4. Gegeben sei das AASS-Programm mit folgender Speicherbelegung:

4	8	12
x	y	z

Zu Beginn seien alle Variablen x, y, z mit dem Wert 1 belegt.

```
1     SET r1 8
2     LOAD r2 [8]
3     LOAD r3 [r1]
4     ADD r2 r2 r3
5     GOLS r3 r2 8
6     ADD r2 r2 r3
7     GOTO 5
8     LOAD r3 [12]
9     ADD r1 r2 r3
10    STORE r1 [12]
11    STORE r2 [12]
12    STOP
```

Welche Werte besitzen die Variablen x, y, z nach der Abarbeitung des Programms?

Aufgabe 8 (Entscheidbarkeit)

1. Nennen sie grundlegende Merkmale von Problemen, die Entscheidbarkeitsprobleme aufweisen.
2. Spezifizieren sie ein Problem, das nicht entscheidbar ist.
3. Spezifizieren sie ein Problem, das entscheidbar ist.
4. Erklären sie, weshalb jede 0-stellige Funktion (Funktion ohne Eingabewert) algorithmisch lösbar ist.