

Bearbeitungszeit: 90 min

Name: _____ **Datum:** _____

Punkte: _____ **Note:** _____

Themenpunkte:

- Laufzeit- und Raumkomplexität
- Deterministische und Nichtdeterministische Algorithmen
- Zahlensysteme

Erlaubte Hilfsmittel:

- Dokumentstecher Stift
- Taschenrechner
- Geodreieck, Lineal
- „Spickzettel“

Nicht erlaubte Hilfsmittel:

- Smartphone, Endgeräte
- Literaturen

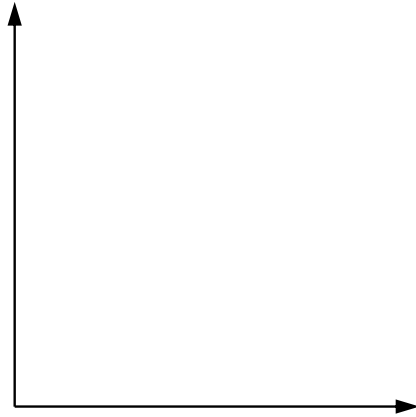
1. Beschreibe den Unterschied zwischen der Raum- und der Laufzeitkomplexität. **(10 Punkte)**

2. Nenne mir drei Merkmale von Nicht-Deterministischen Algorithmen **(6 Punkte)**

3. Nenne mir drei Anwendungsbereiche für deterministische Algorithmen **(6 Punkte)**

4. Was ist eine Permutation in der Informatik und worin liegt die Gefahr? **(6 Punkte)**

5. Beschrifte die Achsen vom Koordinatensystem und stelle die Laufzeitkomplexitäten $O(1)$, $O(\log_2(n))$ und $O(n)$ und $O(n!)$ in Abhängigkeit der Datenmenge n qualitativ dar. Beschrifte die Grafiken. **(5 Punkte)**



6. Überführe die Dezimalzahl 165_{10} in eine Binärzahl **(5Punkte)**

7. Subtrahiere die beiden binären Zahlen 1000001_2 (65_{10}) und 100101_2 (37_{10}) in einem 8-Bit-System **(8 Punkte)**.

8. Wandle die Dezimalzahl -11,625 nach der IEEE 754 in eine Binärdarstellung um

9. Um welchen primitiven Datentyp handelt es sich bei der Aufgabe 8? **(2 Punkte)**

10. Gegeben ist ein Algorithmus (siehe unten), der in der Lage ist Elemente in einer Liste zu generieren und diese Elemente in der Liste aufsteigend zu sortieren. Wie lautet die gesamte Laufzeitkomplexität vom oberen Programmausschnitt? Erkläre die Zusammensetzung der Laufzeitkomplexitäten mathematisch! **(7 Punkte)**

```
import random

def generate_random_list():
    random_list = []
    num_elements = random.randint(1, 100)
    for _ in range(num_elements):
        random_element = random.randint(0, 1000)
        random_list.append(random_element)
    return random_list

def insertion_sort(elements):
    for i in range(1, len(elements)):
        element_to_sort = elements[i]
        j = i
        while j > 0 and element_to_sort < elements[j - 1]:
            elements[j] = elements[j - 1]
            j -= 1
        elements[j] = element_to_sort

elements = generate_random_list()
insertion_sort(elements)
print("Sortierte Liste:", elements)
```

Ich wünsche dir viel Erfolg bei der Bearbeitung!