

Prüfung Algorithmen und Datenstrukturen

Allgemeine Hinweise:

- Laden Sie sich zusätzlich zu dieser Angabe die Textdatei mit dem Titel "loesungsbogen.txt" als Vorlage für Ihre Abgabe herunter, in die Sie Ihre Lösungen zu den Aufgaben der Prüfung eintragen.
- Nutzen Sie bitte ausschließlich die .txt-Datei ("loesungsbogen.txt"), um ihre Lösung abzugeben.
- Die Art und Weise, eine Aufgabe zu lösen, wird ihnen detailliert auf der Angabe beschrieben. Lesen Sie diese daher bitte genau.
- Sie bekommen ab Beginn der Prüfung 180 Minuten Zeit, diese herunterzuladen, zu bearbeiten und Ihre Lösung wieder hochzuladen (alles via Uni2work). Somit haben Sie 90 Minuten extra Zeit, um auf eventuelle technische Probleme reagieren zu können.
- Speichern Sie vor Bearbeitung der Prüfung die Lösungsdatei mit dem Titel "loesungsbogen_[Matrikelnummer].txt" an einem Ort ab, an dem Sie sie jederzeit wiederfinden. Ersetzen Sie dabei [Matrikelnummer] mit ihrer Matrikelnummer.
- Vergessen Sie nicht, die Lösungsdatei möglichst häufig zwischenzuspeichern. So reduzieren Sie Stress durch technische Probleme am Ende der Abgabezeit.
- Notfall-Hotline: 089 / 2180 9306

- Ich habe die Lösung eigenständig und ohne die Hilfe Dritter angefertigt.
- Ich bin der rechtmäßige Nutzer dieses Uni2Work-Accounts und gebe die Lösung nicht für jemand Drittes ab.
- Ich bin zum Zeitpunkt der Prüfung immatrikuliert und kann dies auch durch ein entsprechendes Dokument jederzeit während des Prüfungsprozesses bestätigen.
- Ich veröffentliche keinerlei Prüfungsdokumente wie Aufgabenstellung, Korrektur, etc. im öffentlichen Raum (Online, Aushang, Bekanntenkreis,...).
- Ich Sorge dafür, dass ich regelmäßige Updates meiner Lösungen in Uni2Work mache, sodass es zu verminderten technischen Problemen am Ende kommt. Die letzte innerhalb der Abgabefrist hochgeladene Version wird korrigiert. Achtung: Uni2Work logged Sie nach einer gewissen Zeit der Inaktivität automatisch wieder aus!

Die Prüfung besteht aus 6 Aufgaben. Die Punktezahl ist bei jeder Aufgabe angegeben.

Aufgabe	mögliche Punkte	erreichte Punkte
1.Allgemeine Fragen	22	
2.Graphalgorithmen	26	
3.Sortialgorithmen	16	
4.Algorithmenanalyse	24	
5.Suchbäume	16	
6.Hashing	18	
Summe:	122	
Note:		

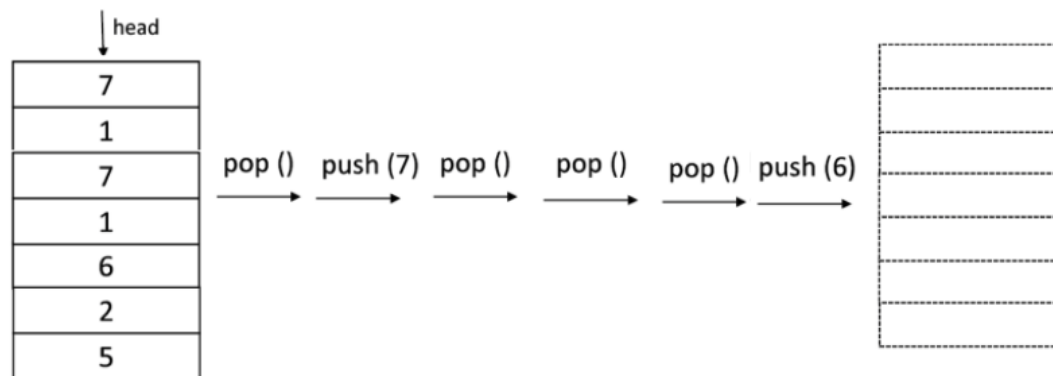
Bewertung Multiple Choice: Geben Sie an, welche Antworten zu folgenden Fragen zutreffen, indem Sie die Ziffern der richtigen Antworten notieren. Alle nicht notierten Ziffern sind automatisch als nicht zutreffend klassifiziert (Sie können sich also nicht enthalten). Beispiel: Angenommen, die Antworten 1,2,3 sind zutreffend, 4,5,6 sind nicht zutreffend, so notieren Sie in der Antwort-Datei 1,2,3 (Antworten 4,5,6 werden *NICHT* notiert und sind damit automatisch als nicht zutreffend gekennzeichnet). Laut Prüfungsordnung bekommen Sie pro gegebener zutreffender Antwort einen Punkt.

Für jede richtige Antwort bekommen Sie einen Punkt, für jede falsche Antwort wird ein Punkt abgezogen. Für jeden Aufgabenteil erhalten Sie nicht weniger als Null Punkte.

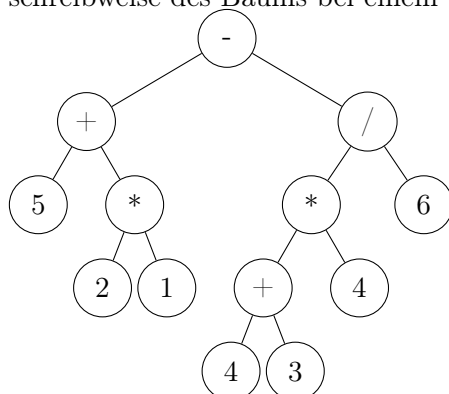
Aufgabe 1 Allgemeine Fragen

(22 Punkte)

- (a) Gegeben sei folgender Stack $\text{head} \rightarrow 7, 1, 7, 1, 6, 2, 5$ (siehe Bild unten). $\text{head} \rightarrow$ zeigt an, welches das oberste Element ist. Welche Aussagen stimmen für das Endergebnis nach den sechs genannten Operationen zu?

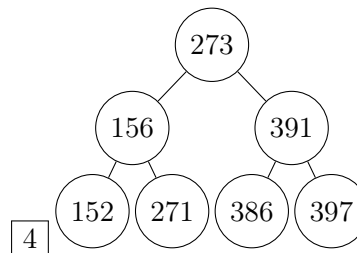
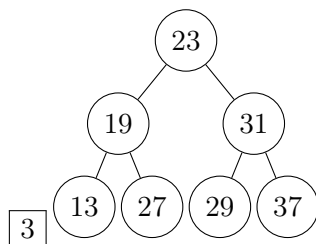
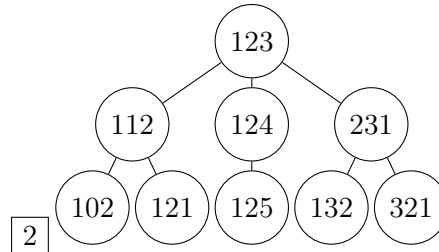
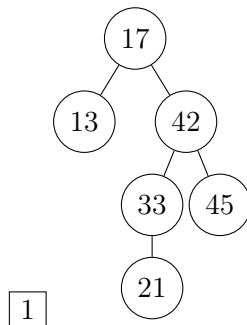


- ☐ 1 Die Nummer 7 steht zweimal drin
 - ☐ 2 Die Nummer 6 steht zweimal drin
 - ☐ 3 Die Länge bleibt unverändert
 - ☐ 4 Die Nummer 1 ist einmal im Stack
 - ☐ 5 Die Nummer 7 ist einmal im Stack
 - ☐ 6 Die Reihenfolge vom Stack ist nicht mehr wichtig
- (b) Gegeben sei der folgende Baum. Welche Vorschläge für die Infix-, Postfix-, und Präfix-schreibweise des Baums bei einem Tiefendurchlauf stimmen?



- ☐ 1 Postfix: $43+4*6/21*5+-$
- ☐ 2 Postfix: $5+21*43+4*6/-$
- ☐ 3 Postfix: $21*5+43+4*6/-$
- ☐ 4 Infix: $5+2*1-4+3*4/6$
- ☐ 5 Praefix: $-+ / 5 * * 6 2 1 + 4 4 3$
- ☐ 6 Praefix: $-+ 5 * 2 1 / * + 4 3 4 6$

(c) Gegeben sind folgende Bäume. Welche davon sind AVL-Bäume?



(d) Welches ist der korrekte Bitvektor für die Menge $M = \{0, 6, 8\}$?

- | | | |
|---|---|-------------------------------|
| 1 | Gegeben das Universum $U = 0, 5, 6, 7, 8$: | $(1, 0, 1, 0, 1)$ |
| 2 | Gegeben das Universum $U = 0, 5, 6, 7, 8$: | $(1, 1, 0, 0, 1)$ |
| 3 | Gegeben das Universum $U = 0, 5, 8, 6, 7, 9$: | $(1, 0, 1, 1, 0, 0)$ |
| 4 | Gegeben das Universum $U = 0, 1, 2, 3, 4, 5, 6, 7, 8$: | $(0, 1, 1, 0, 1, 0, 1, 1, 0)$ |
| 5 | Gegeben das Universum $U = 0, 1, 2, 3, 4, 5, 6, 7, 8$: | $(1, 0, 0, 0, 0, 0, 1, 0, 1)$ |
| 6 | Gegeben das Universum $U = 0, 1, 2, 3, 4, 5, 6, 7, 8$: | $(1, 1, 1, 1, 1, 0, 0, 0, 0)$ |

Aufgabe 2 Graphalgorithmen

(26 Punkte)

- (a) Gegeben sei die folgende Adjazenzliste. Welche Aussagen über den zugehörigen Graphen treffen zu?

$A \rightarrow C \rightarrow D \rightarrow E$

$B \rightarrow A \rightarrow D$

$C \rightarrow A \rightarrow B$

$D \rightarrow A \rightarrow F$

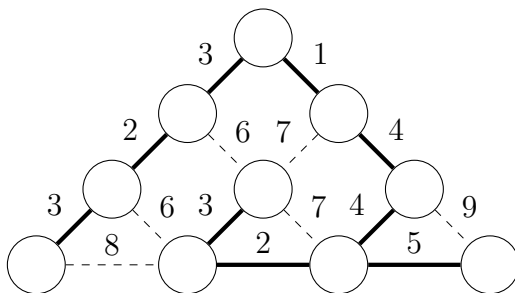
$E \rightarrow C \rightarrow D \rightarrow E$

$F \rightarrow E$

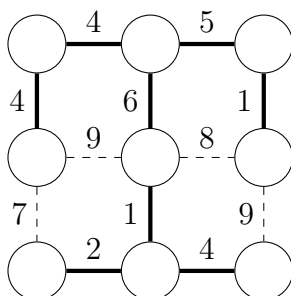
- ☐ 1 Für alle Knotenpaare $(v, w) \in E(G)$ gilt: $v \neq w$.
- ☐ 2 Für alle Knotenpaare existiert ein kürzester Pfad.
- ☐ 3 Zwischen allen Knotenpaaren existiert ein Pfad.
- ☐ 4 Der Graph ist ein Baum.
- ☐ 5 Der Graph ist planar (lässt sich zeichnen, ohne dass sich Kanten kreuzen).
- ☐ 6 Die Liste repräsentiert einen ungerichteten Graphen.
- ☐ 7 Genau ein Knoten hat Knotengrad 3.
- ☐ 8 Jeder Pfad im Graphen ist einfach.

- (b) Die folgende Illustrationen enthalten Graphen mit gestrichelten und durchgezogenen Kanten. Welche der durchgezogenen Kanten repräsentieren valide minimale Spannbäume dieser Graphen?

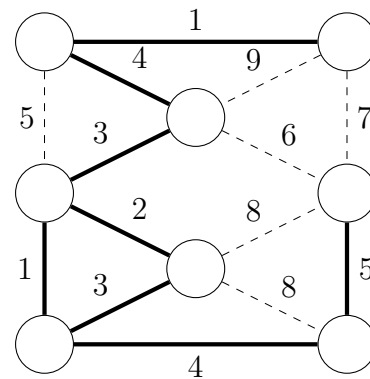
☐ 1



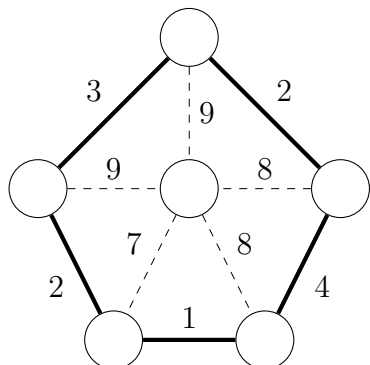
☐ 3



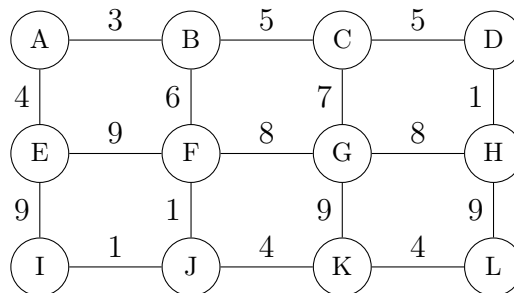
☐ 2



☐ 4

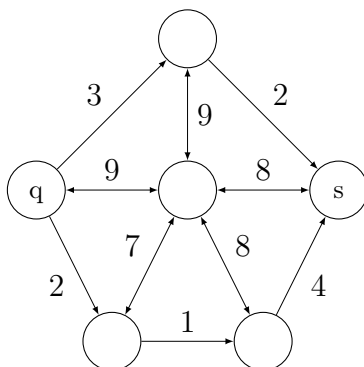


- (c) Wenden Sie den Dijkstra-Algorithmus mit Startknoten A auf folgenden Graphen G an. Welche Aussagen treffen zu?



- ☐ 1 D wird vor K bearbeitet.
- ☐ 2 Der kürzeste Weg zu K hat Länge 18.
- ☐ 3 Es gibt genau zwei Knoten, die die gleiche Distanz zu A haben.
- ☐ 4 Der kürzeste Weg zu H hat Länge 23.
- ☐ 5 C wird direkt nach B bearbeitet.
- ☐ 6 Die Bearbeitungsreihenfolge von H und K lässt sich vertauschen.
- ☐ 7 Die Bearbeitung von G verbessert keine bisher gefundenen Wege.
- ☐ 8 Der kürzeste Pfad von A nach E hat Distanz 9.
- ☐ 9 L wird zuletzt bearbeitet.
- ☐ 10 Der kürzeste Pfad von A nach E hat Distanz 8.

- (d) Gegeben folgendes Flussnetzwerk mit Quelle q und Senke s , welche Aussagen sind korrekt?



- ☐ 1 Der maximale Fluss ist 11.
- ☐ 2 Der maximale Fluss ist 12.
- ☐ 3 Der maximale Fluss ist 13.
- ☐ 4 Der maximale Fluss ist 14.

Aufgabe 3 Sortieralgorithmen

(16 Punkte)

- (a) Gegeben sei das folgende Array
[17, 7, 9, 11, 4, 1, 13, 2, 20, 14, 19, 3, 16, 6, 15, 18, 10, 8, 5, 12].

und folgende Zwischenergebnisse eines einfachen Sortieralgorithmus. Welcher Algorithmus aus der Vorlesung wurde jeweils benutzt?

- i. [1, 2, 3, 4, 7, 9, 11, 13, 14, 17, 19, 20, 16, 6, 15, 18, 10, 8, 5, 12]

- ☐ 1 MergeSort
- ☐ 2 BubbleSort
- ☐ 3 InsertionSort
- ☐ 4 SelectionSort

- ii. [1, 2, 3, 4, 5, 6, 7, 8, 9, 14, 19, 20, 16, 17, 15, 18, 10, 13, 11, 12]

- ☐ 1 MergeSort
- ☐ 2 BubbleSort
- ☐ 3 InsertionSort
- ☐ 4 SelectionSort

- iii. [4, 1, 7, 2, 9, 11, 13, 3, 14, 6, 15, 16, 10, 9, 5, 12, 17, 18, 19, 20]

- ☐ 1 MergeSort
- ☐ 2 BubbleSort
- ☐ 3 InsertionSort
- ☐ 4 SelectionSort

- (b) Ein Array wird schrittweise mit einem vergleichsbasierten Sortieralgorithmus der Vorlesung sortiert:

[17, 7, 9, 11, 4, 1, 13, 2, 20, 14, 19, 3, 16, 6, 15, 18, 10, 8, 5, 12]

1. Schritt: [7, 9, 11, 4, 1, 2, 3, 6, 10, 8, 5, 12, 17, 13, 20, 14, 19, 16, 15, 18]
2. Schritt: [4, 1, 2, 3, 5, 7, 9, 11, 6, 10, 8, 12, 17, 13, 14, 16, 15, 18, 20, 19]
3. Schritt: [1, 2, 3, 4, 5, 7, 6, 8, 9, 11, 10, 12, 13, 14, 15, 17, 16, 18, 19, 20]
4. Schritt: [1, 2, 3, 4, 5, 7, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
5. Schritt: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]

Was ist die Durchschnitts-Laufzeit dieses Algorithmus bei einem Array der Länge n ?

- ☐ 1 $O(n^{-2})$
- ☐ 2 $O(n)$
- ☐ 3 $O(n \cdot \log(n))$
- ☐ 4 $O(\log(n))$

Aufgabe 4 Algorithmenanalyse

(24 Punkte)

Die folgenden Algorithmen wurden erstellt, um das Maximum einer Liste von Zahlen zu bestimmen. Sie dürfen annehmen, dass nur nicht-leere Listen mit positiven ganzen Zahlen übergeben werden. $\text{max}()$ ist in dieser Aufgabe eine vordefinierte Funktion, die in $\mathcal{O}(1)$ das Maximum zweier Zahlen bestimmt. Zusätzlich wird ein Algorithmus als *total korrekt* charakterisiert, wenn er partiell korrekt ist **und** terminiert.

(a) Welche Aussagen treffen auf folgenden Algorithmus zu?

```
def max1(liste):
    n = length(liste)
    if n == 1:
        return liste[0]
    if n == 2:
        return max(liste[0], liste[1])
    return max(max1(liste[0, ..., n/2]), max1(liste[n/2+1, ..., n]))
```

- ☐ 1 max1 hat im Worst-Case eine Komplexität von $\mathcal{O}(\log n)$
- ☐ 2 max1 hat im Worst-Case eine Komplexität von $\mathcal{O}(n^2)$
- ☐ 3 max1 hat im Worst-Case eine Komplexität von $\mathcal{O}(n)$
- ☐ 4 max1 ist partiell korrekt.
- ☐ 5 max1 ist determiniert.
- ☐ 6 max1 ist total korrekt.

(b) Welche Aussagen treffen auf folgenden Algorithmus zu?

```
def max2(liste):
    n = length(liste)
    for i in 0...n-1:
        for j in 0...n-1:
            if liste[i] < liste[j]:
                liste[i], liste[j] = liste[j], liste[i]
    return liste[n-1]
```

- ☐ 1 max2 hat im Worst-Case eine Komplexität von $\mathcal{O}(\log n)$
- ☐ 2 max2 hat im Worst-Case eine Komplexität von $\mathcal{O}(n^2)$
- ☐ 3 max2 hat im Worst-Case eine Komplexität von $\mathcal{O}(n)$
- ☐ 4 max2 terminiert.
- ☐ 5 max2 ist determiniert.
- ☐ 6 max2 ist total korrekt.

(c) Welche Aussagen treffen auf folgenden Algorithmus zu?

```
def max3(liste):  
    res = liste[0]  
    for i in 0...length(liste)-1:  
        res = max(liste[i], res)  
    return res
```

- ☐ 1 max3 hat im Worst-Case eine Komplexität von $\mathcal{O}(n)$
- ☐ 2 max3 hat im Worst-Case eine Komplexität von $\mathcal{O}(\log n)$
- ☐ 3 max3 hat im Worst-Case eine Komplexität von $\mathcal{O}(1)$
- ☐ 4 max3 ist deterministisch.
- ☐ 5 max3 ist partiell korrekt.
- ☐ 6 max3 ist endrekursiv.

(d) Welche Aussagen treffen auf folgenden Algorithmus zu?

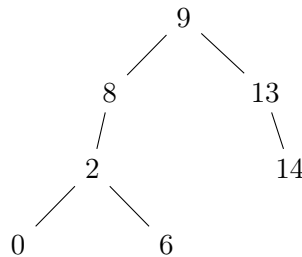
```
def max4(liste):  
    pick = random_choice(liste) // waehle zufaellig Element aus  
    for it in 0...1000:  
        pick = max(pick, random_choice(liste))  
    return pick
```

- ☐ 1 max4 hat im Worst-Case eine Komplexität von $\mathcal{O}(n)$
- ☐ 2 max4 hat im Worst-Case eine Komplexität von $\mathcal{O}(\log n)$
- ☐ 3 max4 hat im Worst-Case eine Komplexität von $\mathcal{O}(1)$
- ☐ 4 max4 ist deterministisch.
- ☐ 5 max4 ist terminiert.
- ☐ 6 max4 ist total korrekt.

Aufgabe 5 Suchbäume

(16 Punkte)

- (a) Kreuzen Sie die gültige(n) Reihenfolge(n) an, in der die Werte eingefügt werden müssen, damit der unten abgebildete binärer Suchbaum entsteht. Starten Sie dabei mit einem leeren Baum.

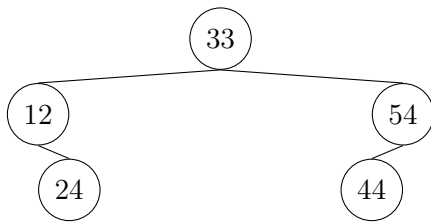


- ☐ 1 9, 8, 2, 14, 0, 13, 6
- ☐ 2 9, 13, 8, 2, 14, 6, 0
- ☐ 3 9, 8, 2, 13, 0, 14, 6
- ☐ 4 9, 13, 14, 8, 2, 6, 0
- ☐ 5 9, 13, 2, 8, 0, 6, 14
- ☐ 6 9, 8, 14, 13, 2, 6, 0

- (b) Gegeben sind Suchverläufe (besuchte Knoten) bei einer Suche in einem binären Suchbaum, der ganze Zahlen zwischen 1 und 100 speichert. Nachdem die angegebenen Schlüssel traversiert wurden, wurde die Suche erfolglos abgebrochen. Kreuzen Sie die Menge möglicher Schlüsselwerte (zwischen 1 und 100) an, nach denen gesucht worden sein könnte.

- ☐ 1 Besuchte Knoten: 14, 38, 19, 15, 17 Gesuchte Schlüsselwerte: 16, 18
- ☐ 2 Besuchte Knoten: 14, 38, 19, 15, 17 Gesuchte Schlüsselwerte: 14, 16
- ☐ 3 Besuchte Knoten: 14, 38, 19, 15, 17 Gesuchte Schlüsselwerte: 10, 11
- ☐ 4 Besuchte Knoten: 82, 3, 12, 9, 4 Gesuchte Schlüsselwerte: 5, 6
- ☐ 5 Besuchte Knoten: 82, 3, 12, 9, 4 Gesuchte Schlüsselwerte: 7, 8
- ☐ 6 Besuchte Knoten: 82, 3, 12, 9, 4 Gesuchte Schlüsselwerte: 10, 11

(c) Gegeben sei der folgende balancierte AVL-Baum.



Löschen Sie nun die Knoten 24, 44 und 54 und fügen Sie die 8, 4 und die 2 hinzu. Kreuzen Sie alle verwendeten Rotationsarten an, die vorgenommen werden müssen, um den Baum zu balancieren.

☐ 1 LR-Rotation

☐ 2 RL-Rotation

☐ 3 LL-Rotation

☐ 4 RR-Rotation

Aufgabe 6 Hashing

(18 Punkte)

- (a) Welche der folgenden Hash-Funktionen verteilt Schlüssel im Bereich 0-1000 auf zehn Positionen 0 bis 9 so gleichverteilt wie möglich?

☐ 1 $h_1(x) = 4x \mod 10$

☐ 2 $h_2(x) = x^2 \mod 10$

☐ 3 $h_3(x) = x^3 \mod 10$

☐ 4 $h_4(x) = 3x^2 \mod 10$

- (b) Für welche Listen ganzer Zahlen ist die Hashfunktion $h(x) = x^2 \mod 5$ perfekt?

☐ 1 6, 3, 8, 11, 1

☐ 2 3, 14, 6, 7, 0

☐ 3 13, 5, 9, 12, 6

☐ 4 5, 9, 7, 12, 3

- (c) Gegeben sei die Eingabe 94, 75, 68, 61, 90, 48, 70, 58, 21, 23. Fügen Sie diese Werte in eine Hashtabelle der Größe 10 mit Positionen 0 bis 9 mittels geschlossenem Hashing mit offener Adressierung ein. Verwenden Sie die Hashfunktion $h(x) = x^5 \mod 10$ und nutzen Sie lineares Sondieren ($c=1$) bei Kollisionen. Welche Aussagen treffen zu?

☐ 1 Alle Schlüssel können in die Hashtabelle eingefügt werden.

☐ 2 90 erzeugt die erste Kollision.

☐ 3 70 wird bei Position 2 eingefügt.

☐ 4 58 wird bei Position 8 eingefügt.

☐ 5 23 wird nach 5 Sondierungen bei Position 9 eingefügt.

☐ 6 21 wird nach 5 Sondierungen bei Position 6 eingefügt.

- (d) Nutzen Sie wieder die Eingabe 94, 75, 68, 61, 90, 48, 70, 58, 21, 23. Die Hashtabelle hat wieder Positionen 0 bis 9. Ebenfalls nutzen Sie die Hashfunktion $h(x) = x^5 \mod 10$. Hashen Sie aber diesmal mit offenem Hashing mit geschlossener Adressierung. Welche Aussagen treffen zu?

☐ 1 Alle Positionen werden mit Schlüsseln belegt.

☐ 2 Die Positionen 2, 6, 7 und 9 enthalten keine Schlüssel.

☐ 3 58 muss zweimal sondiert werden.

☐ 4 Position 1 enthält drei Schlüssel.