

2. Zwischentest zur Vorlesung Algorithmen und Datenstrukturen (Wintersemester 2021/2022)

Bitte beachten Sie die folgenden Hinweise!

1. **Personalien:** Bitte tragen Sie Ihre Daten gut lesbar ein.

Name	
Vorname	
Matrikelnummer	
Übungsgruppe	

2. **Papier:** Tragen Sie Ihre Lösungen direkt unterhalb der Aufgaben ein.
Verwenden Sie kein eigenes Papier.
3. Sie dürfen ein einseitig handgeschriebenes DIN-A4-Blatt mit Notizen verwenden. **Weitere Hilfsmittel sind nicht zugelassen.**
4. Schreiben Sie **nicht** mit Bleistift oder einem roten Stift.
5. Die Bearbeitungszeit beträgt 60 Minuten.

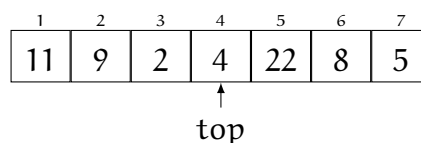
Wir wünschen Ihnen viel Erfolg!

Aufgabe	1	2	3	4	5	6	Gesamt
Punkte	15	2	16	10	7	10	60
Ergebnis							

Aufgabe 1 — Datenstrukturen

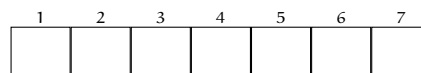
- (a) Gegeben sei dieser Stapel, der wie in der Vorlesung mit einem Feld implementiert wurde:

/ 3 P.



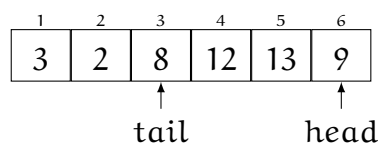
Wie sieht der Stapel aus, nachdem folgende Operationen in dieser Reihenfolge darauf angewendet wurden? Zeichnen Sie auch den Zeiger top ein.

1. Push(3)
2. Top()
3. Push(18)
4. Pop()



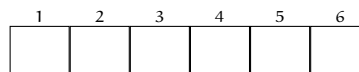
- (b) Gegeben sei diese Schlange, die wie in der Vorlesung mit einem Feld implementiert wurde:

/ 3 P.



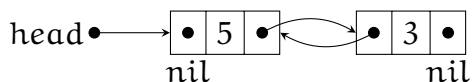
Wie sieht die Schlange aus, nachdem folgende Operationen in dieser Reihenfolge darauf angewendet wurden? Zeichnen Sie auch die Zeiger head und tail ein.

1. Dequeue()
2. Enqueue(6)



- (c) Gegeben sei diese Liste, die wie in der Vorlesung implementiert wurde:

/ 3 P.



Wie sieht die Liste (in schematischer Darstellung) aus, nachdem folgende Operationen in dieser Reihenfolge darauf angewendet wurden? Zeichnen Sie die Liste mit allen Zeigern (einschließlich head und nil) hierunter.

1. Delete(Search(5))
2. Insert(2)
3. Insert(4)

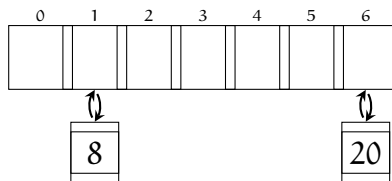
Name:

Matrikelnummer:

- (d) Gegeben sei die folgende Hashtabelle mit Verkettung und folgende Hashfunktion:
 $h : k \mapsto k \bmod 7$. Ergänzen Sie die folgenden Schlüssel, die in dieser Reihenfolge mittels h zugefügt werden.

/ 3 P.

1. 74
2. 25
3. 28



- (e) Gegeben sei die folgende Hashtabelle mit offener Adressierung und linearem Sondieren und die Hashfunktion h aus Teilaufgabe (d). Fügen Sie die folgenden Schlüssel in dieser Reihenfolge in die Tabelle ein. Zeichnen Sie gegebenenfalls die Sondierfolge durch Pfeile ein.

/ 3 P.

1. 13
2. 68

0	1	2	3	4	5	6
		2			33	

Aufgabe 2 — Median

/ 2 P.

Gegeben sei ein aufsteigend sortiertes Feld $\text{int}[]$ F und zwei Indizes ℓ, r , wobei $1 \leq \ell \leq r \leq F.\text{length}$. Wie können Sie den Median des Teilfeldes $F[\ell..r]$ in konstanter Zeit berechnen?

Aufgabe 3 — Zufallssumme

Gegeben sei der folgende Algorithmus `Alg01`. Darin gebe `Random( $\ell$ )` in konstanter Zeit zufällig gleichverteilt eine Zahl aus $\{1, 2, \dots, \ell\}$ zurück. Sie können annehmen, dass $0 \leq k \leq \ell$ gilt.

```
Alg01(int k, int  $\ell$ ):  
  A = new int[1.. $\ell$  + 1]  
  for i = 1 to  $\ell$  do  
    A[i] = Random( $\ell$ )  
  sum = 0  
  for i = 1 to  $\ell$  do  
    if A[i]  $\leq$  k then  
      sum = sum + A[i]  
  return sum
```

- (a) Die Laufzeit von `Alg01` liegt in $\Theta(\text{_____})$ in Abhängigkeit von k und ℓ . / 1 P.
- (b) Wenn man $\ell = 0$ wählt, gibt `Alg01` _____ zurück. / 1 P.
- (c) Wenn man $k = 0$ wählt, gibt `Alg01` _____ zurück. / 1 P.
- (d) Wenn man $k = 1$ und $\ell = 1$ wählt, gibt `Alg01` _____ zurück. / 1 P.
- (e) Wenn man $k = 1$ und $\ell = 2$ wählt, ist der **größtmögliche** Rückgabewert von `Alg01` _____ / 1 P.
- (f) Wenn man $k = 1$ und $\ell = 2$ wählt, ist der **kleinstmögliche** Rückgabewert von `Alg01` _____ / 1 P.
- (g) Wenn man $k = 1$ und $\ell = 2$ wählt, ist der **erwartete** Rückgabewert von `Alg01` _____ / 1 P.
- (h) Wenn man $k = 2$ und $\ell = 2$ wählt, ist der **erwartete** Rückgabewert von `Alg01` _____ / 1 P.

Name:

Matrikelnummer:

- (i) Sei für alle $i \in \{1, \dots, \ell\}$ die Indikatorzufallsvariable $X_i = \begin{cases} 1, & \text{falls } A[i] \leq k \\ 0, & \text{sonst} \end{cases}$

/ 1 P.

Die Wahrscheinlichkeit, dass $X_i = 1$ ist, ist in Abhängigkeit von k und ℓ gleich

$P[X_i = 1] = \underline{\hspace{2cm}}$.

- (j) Wir definieren für alle $i \in \{1, \dots, \ell\}$ die Zufallsvariable $Y_i = \begin{cases} A[i], & \text{falls } X_i = 1 \\ 0, & \text{sonst} \end{cases}$

/ 3 P.

Was ist der Erwartungswert von Y_i in Abhängigkeit von k und ℓ (er soll **nicht** von $A[i]$ abhängen)?

$E[Y_i] =$

- (k) Definieren Sie eine Zufallsvariable Y , die den Rückgabewert von `Alg01` angibt, in Abhängigkeit von Y_i für $i \in \{1, \dots, \ell\}$.

/ 1 P.

- (l) Was ist der Erwartungswert von Y in Abhängigkeit von k und ℓ ?

/ 2 P.

$E[Y] =$

- (m) In einem Experiment wurde `Alg01(42, 50)` genau 1000 mal aufgerufen und der durchschnittliche Rückgabewert berechnet. Kann man damit rechnen den durchschnittlichen Rückgabewert zu erhöhen, wenn man stattdessen `Alg01(42, 1000000)` 1000 mal aufruft? Begründen Sie Ihre Antwort.

/ 1 P.

Aufgabe 4 — Sortiervverfahren

- (a) Nennen Sie zwei aus der Vorlesung bekannte Sortiervverfahren, die *stabil* sind.

/ 2 P.

- (b) Geben Sie ein Sortiervverfahren an, das sich besonders gut eignet, um eine Menge von n Personen im Alter zwischen 0 und 140 Jahren nach ihrem genauen Alter (in Tagen) zu sortieren. Sie dürfen annehmen, dass jedes Alter gleich wahrscheinlich ist. Geben Sie die (ggf. erwartete) Laufzeit ihres Verfahrens an und begründen Sie Ihre Antwort.

/ 4 P.

Verfahren: _____ **Laufzeit:** Θ (_____)

Begründung der Laufzeit:

- (c) Geben Sie ein Sortiervverfahren an, das sich besonders gut eignet, um eine Menge von n Primzahlen zu sortieren. Geben Sie die (ggf. erwartete) Laufzeit ihres Verfahrens an und begründen Sie Ihre Antwort.

/ 4 P.

Verfahren: _____ **Laufzeit:** Θ (_____)

Begründung der Laufzeit:

Aufgabe 5 — Hashing

In der Vorlesung haben wir für Hashing mit offener Adressierung das doppelte Hashing mittels einer Hashfunktion vom Typ

$$h(k, i) = (h_0(k) + i \cdot h_1(k)) \mod m$$

kennengelernt, wobei $h_1(k)$ und m für alle Schlüssel k teilerfremd sein mussten. Um ein etwas einfacheres Verfahren zu verwenden, wird der Vorschlag gemacht, die folgende Hashfunktion h^* zu verwenden, welche ebenfalls zwei Hashfunktionen h_0 und h_2 nutzt:

$$h^*(k, i) = (h_0(k) + h_2(i)) \mod m$$

- (a) Welche Voraussetzungen müssen für h_2 und m gelten, damit h^* für jedes k eine gültige Sondierfolge berechnet?

/ 3 P.

- (b) Bewerten Sie den Vorschlag h^* zu verwenden, indem Sie ihn mit den aus der Vorlesung bekannten Sondiermethoden vergleichen.

/ 4 P.

Aufgabe 6 — Binäre Suchbäume

/ 10 P.

Gegeben sei ein binärer Suchbaum T der Höhe h , wie wir ihn in der Vorlesung kennengelernt haben. In T sind die Schlüssel aller Knoten unterschiedlich und ganzzahlig.

Geben Sie eine Methode in Pseudocode an, mit der alle Schlüssel in T , die im Intervall $I = \{a, \dots, b\}$ liegen, ausgegeben werden. Verwenden Sie hier zum Ausgeben des Schlüssels eines Knotens z innerhalb Ihres Codes `print(z.key)`. Die Laufzeit Ihres Algorithmus sollte in $O((k + 1) \cdot (h + 1))$ liegen, wobei k die Anzahl der Schlüssel aus T in I ist. Nach Ausführung des Algorithmus soll T die gleichen Schlüssel enthalten wie zuvor und immer noch ein binärer Suchbaum sein. Beachten Sie auch, dass T leer sein kann. Begründen Sie die **Korrektheit** und **Laufzeit** Ihres Algorithmus kurz.

`DruckeSchlüsselInIntervall(BinarySearchTree T, int a, int b):`

Name:

Matrikelnummer:
