

Fragenkatalog für Studierende ADS Mündliche Prüfung HS24/25

Fragen
Grundlagen
Angenommen, während der Ausführung eines Java-Programms kommt es zur Ausführung der Anweisung $x = x + 1$. Angenommen weiter, x sei eine int-Variable und diese Anweisung stehe unter der Vorbedingung $x = 5$. Nennen Sie eine möglichst genaue Nachbedingung (Invariante).
Schreiben Sie einen Algorithmus in Pseudo-Code, der überprüft, ob eine Zahl eine Primzahl ist. Tipp: Verwenden Sie die Modulo-Funktion.
Erklären Sie wie binäre Suche funktioniert.
Wie ist die Worst-Case-Laufzeit von binärer Suche? Wieso?
Was ist ein Stack; welches sind die grundlegenden Operationen; wie kann er implementiert werden?
Was ist eine Queue; welches sind die grundlegenden Operationen; wie kann sie implementiert werden; was ist eine PriorityQueue und wie funktioniert sie?
Erläutern Sie Unterschiede und Gemeinsamkeiten zwischen Queues und Stacks.
Es wird eine Datenstruktur benötigt, welche variable Grösse hat und eine schnelle Suche nach Elementen basierend auf einer bekannten Eigenschaft ermöglicht. Die Elemente müssen nicht sortiert werden. Welche Datenstruktur wäre am besten geeignet und warum?
Laufzeit
Welche Laufzeit hat folgender Code - grobe Abschätzung genügt?
1. Was hält die O-Notation (Big-O) fest und 2. wie lautet die Definition der O-Notation?
Nennen Sie einige typische Komplexitätsklassen und ordnen Sie diese nach Aufwand.
Nennen Sie für vier Komplexitätsklassen (O-Notation) einen Algorithmus mit dieser Laufzeit.
Welche Rechenregeln gelten bei der Landau-Notation? (z.B. $O(n^2 + n^3) \rightarrow O(n^3)$)
Welche Laufzeit hat folgender Code (Beispiel 1)?
Welche Laufzeit hat folgender Code (Beispiel 2)?
Welche Laufzeit hat die Suche in einer unsortierten LinkedList? Was ist, wenn die Liste sortiert ist?
Was ist der Unterschied zwischen Average und Worst Case Laufzeit?
Falls immer $f(x) < g(x)$ ist, ist dann $f = O(g)$?
Hat folgender Algorithmus Laufzeit $O(n^3)$?
Welche Laufzeit hat folgender Code?
Was sagt das Amdahl-Gesetz aus?
Rekursion
Warum sind iterative Algorithmen oft schneller als ihre rekursiven Varianten?
Skizzieren Sie eine rekursive Methode, welche die Elemente einer einfach verketteten Liste in umgekehrter Reihenfolge ausgibt (in Pseudocode).
Was ist eine Endrekursion?
Definieren Sie den Datentyp "Liste" rekursiv.
Wie lautet die Definition der Fibonacci-Zahlen?
Wie kann man die Fibobacci-Zahlen iterativ berechnen?
Nennen Sie ein Beispiel rekursiver Programmierung aus der Vorlesung.
Wie kann man die Fibonacci-Zahlen rekursiv berechnen? Was passiert dabei mit der Laufzeit
Ist folgender Code endrekursiv (Beispiel 1)?
Ist folgender Code endrekursiv (Beispiel 2)?
Aus der Definition der Fakultät $n! = 1 \times 2 \times \dots \times n$ kann man die Beziehung $n! = n(n - 1)!$ herleiten. Auf der Basis dieser Beziehung sollte die folgende Funktion zu einem gegebenen $n \geq 0$ die Fakultät $n!$ berechnen. Irgendetwas ist aber schief gelaufen. Was ist es und wie muss man das korrigieren?
Listen
Erklären Sie, was eine Liste ist und welches die Grundoperationen sind.
Erläutern Sie Unterschiede und Gemeinsamkeiten zwischen Queues und Lists.

Nennen Sie eine Operation, die typischerweise in Java in einer LinkedList schneller geht als in einer ArrayList. Wieso?
Wie kann man in einer einfach verketteten Liste ein Element an Position i einfügen?
Was ist der Unterschied zwischen einer einfach und einer doppelt verketteten Liste?
Was sind die wichtigsten Unterschiede zwischen einer einfachen und einer doppelten Linked List?
Erläutern Sie Unterschiede und Gemeinsamkeiten zwischen Stacks und Lists.
Welche Laufzeit hat das Löschen eines Elementes in einer doppelt verketteten Liste?
Warum ist ein "tail"-pointer bei Listen oft hilfreich?
Gegeben ist eine einfach-verkettete Liste OHNE Tail-Pointer. Welche Datenstruktur sollte man damit NICHT implementieren, einen Stack oder eine Queue?
Welche Laufzeit hat das Einfügen eines Elements an der ersten Position in einer ArrayList?
Nennen Sie eine Operation, die typischerweise in Java in einer ArrayList schneller ist als in einer LinkedList
Welche Laufzeit hat das Einfügen eines Elements an der ersten Position in einer einfach-verketteten Liste?
Nennen Sie zwei Implementierungen von Listen in Java
Welche Methoden hat ein Java-Iterator?
Wie verwendet man einen Java-Iterator?
Was ist in Java der Unterschied zwischen einer LinkedList und einer ArrayList? Was bedeutet das für die Performance.
Gegeben sei ein unbekannter ADT mit zwei Operationen void foo(int x) und int bar(). Sie rufen nun foo() und bar() nacheinander in folgender Reihenfolge auf: <pre>foo(1); foo(2); System.out.println(bar()); // Prints "2" foo(3); foo(4); System.out.println(bar()); // Prints "4" System.out.println(bar()); // Prints "3" System.out.println(bar()); // Prints "1"</pre> Handelt es sich bei dem ADT um einen Stack, eine Queue oder etwas anderes? Konkrete Abfolge der foo()- und bar()-Operationen (und natürlich die richtige Antwort!) kann in der Prüfung abweichen.
Bäume
Kann man aus der Pre- und Postorder-Reihenfolge einen Baum immer eindeutig wiederherstellen?
Was ist ein binärer Baum?
Was ist ein binärer Suchbaum?
Wo findet man in einem binären Suchbaum den grössten Knoten?
Ist das ein Baum? Begründen Sie!
Welche Traversierungsarten bei Bäumen kennen Sie?
Es wird eine Datenstruktur benötigt, die eine schnelle Suche nach Elementen basierend auf einer bekannten Eigenschaft ermöglicht. Die Elemente müssen eine natürliche Reihenfolge einhalten. Welche Datenstruktur wäre am besten geeignet und warum?
Fügen Sie die Zahlen der Reihe nach in einen binären Suchbaum ein. Was erkennen Sie?
Wie hoch ist ein binärer Baum mit n Elementen mindestens und höchstens?
Skizzieren Sie einen vollständig gefüllten, binären Suchbaum der Höhe 3 und löschen Sie anschliessend den gezeigten Knoten aus dem Baum.
Worin unterscheiden sich Implementierungen von pre- in- und postorder?
Was passiert, wenn Sie die Knoten eines binären Suchbaums inorder ausgeben?
In einem bin. Suchbaum mit ungerade vielen Elementen ($n = 2k+1$) steht an jedem Knoten, wie viele Knoten der gesamte Teilbaum hat. Wie kann man mit dieser Information den Median bestimmen?

Balancierte Bäume

Skizzieren Sie das Problem von binären Suchbäumen, das balancierte Bäume beheben.

Welche Operationen werden benötigt, um einen durch Einfügen unbalancierten AVL-Baum wieder zu balancieren? Beschreiben Sie diese anhand einer Skizze.

Warum hat das Einfügen/Suchen in AVL-Bäumen logarithmische Laufzeit?

Wie lautet die Definition eines AVL-Baums?

Welches ist die wichtigste Eigenschaft, die AVL-Bäume erfüllen müssen?

Erklären Sie, was ein AVL-Baum ist und welche die wichtigsten Operationen sind.

Was sind die Hauptunterschiede zwischen Stacks und AVL-Bäumen?

Welche Komplexität hat das Einfügen in einen AVL-Baum?

Welche Komplexität hat eine Rotation in einem AVL-Baum?

Ein AVL-Baum sei durch Einfügen eines Elements aus der Balance geraten. Wie kann man erkennen, ob eine einfache oder eine doppelte Rotation gebraucht wird, um die Balance wiederherzustellen?

Erklären Sie was ein 2-3-4 Baum ist.

Erklären Sie was ein Rot-Schwarz-Baum ist.

Welche Java-Klassen verwenden Rot-Schwarz-Bäume?

B-Bäume

Was macht man, wenn ein Knoten in einem B-Baum "überläuft"?

Was ist ein B-Baum der Ordnung k ?

Wie wird die Höhe eines B-Baumes erhöht?

Wie funktioniert Einfügen in einem B-Baum?

Wie viele Knoten (Pages) hat ein B-Baum der Ordnung k und Höhe h höchstens?

Wofür werden B-Bäume in der Praxis eingesetzt?

Graphen

Sie haben einen ungerichteten Graphen mit vorgegebenen Kanten. Finden Sie eine geschlossene Tour (Anfangsknoten = Zielknoten), bei der jede Kante genau einmal genommen wird, oder begründen Sie, warum es keine solche Tour geben kann.

Was ist Breitensuche? Was ist Tiefensuche?

Zählen Sie ein paar Eigenschaften von Grafen auf.

Wie viele Kanten hat ein vollständiger (kompletter) ungerichteter Graph mit n Knoten?

Was berechnet der Algorithmus von Dijkstra? Wie funktioniert er?

Wieso ist das ein Beispiel für einen Greedy-Algorithmus?

Was ist eine Adjazenzmatrix?

Was ist eine Adjazenzliste?

Betrachten Sie den gerichteten gewichteten Graphen (Abbildung 2). Bestimmen Sie den maximalen Fluss von A nach H.

Was bestimmt der Algorithmus von Ford und Fulkerson und wie funktioniert er?

Gegeben sei der gerichtete Graph aus Abbildung 2. Wie lautet die Reihenfolge der Knoten beim Durchlaufen des Graphen in Breitensuche, wenn man beim Knoten A beginnt?

Welche Graphenrepräsentation würden Sie verwenden, um einen Graphen als Textfile zu speichern?

Wie lautet die Definition eines ungerichteten (gerichteten, gewichteten) Graphen?

Wie kann man einen ungerichteten Graphen als gerichteten Graphen repräsentieren?

Was ist topologisches Sortieren?

Was ist das Traveling Salesman Problem? Was ist der Aufwand zur Bestimmung des kürzesten Weges und wie könnte man das Problem näherungsweise lösen?

Sortieren

Wie funktioniert Bubblesort? Was ist die Laufzeit, und warum?

Wie funktioniert Insertionsort?

Wie funktioniert Selectionsort?

Wie ist die Laufzeit von BubbleSort bei absteigend oder aufsteigend sortierter Folge?

Eine Folge von Zahlen die aufsteigend sortiert werden soll, ist bereits absteigend sortiert. Wie viele Vergleiche und Vertauschungen brauchen Bubblesort und Selectionsort?

Bei Distribution Sort kann es u.U. notwendig sein, die Schlüssel noch auf Array-Indizes abzubilden. Beim Hashing ergibt sich dasselbe Problem. Trotzdem kann man bei Distribution Sort keine Hashfunktion verwenden, um das Problem zu lösen. Wieso nicht?
Welche Eigenschaften hat Selectionsort? - stabil (Sortierung bleibt erhalten)? - in situ (zusätzliche Speicherbedarf ist unabhängig von der Anzahl der zu sortierenden Elemente)? - Laufzeit?
Wie funktioniert Quicksort?
Warum ist die Worst-Case-Laufzeit von Quicksort quadratisch?
Wie beeinflusst bei Quicksort die Wahl des Pivotelements das Laufzeitverhalten, wenn das Array schon sortiert ist?
Welches Charakteristikum von Quicksort ist dafür verantwortlich, dass seine Worst-Case-Performance nicht besser sein kann als $O(n \log n)$? Wieso ist es überhaupt möglich, dass Distribution Sort eine Chance auf eine bessere Performance als $O(n \log n)$ hat?
Welchen Sortieralgorithmus würden Sie nehmen, wenn Sie die Einträge in einer Datei sortieren wollen, die nicht komplett in den Hauptspeicher passt?
Was bedeutet es, wenn ein Sortieralgorithmus stabil ist?
Wie funktioniert der interne Merge-Sort-Algorithmus mit einem Array? Welches ist der Aufwand?
Algorithmenmuster
Das acht-Damen-Problem bezeichnet das Problem, acht Damen auf einem Schachbrett so anzuordnen, dass sich je zwei Damen nicht bedrohen. Beschreiben Sie, wie Sie dieses Problem mit Backtracking lösen würden. (Hinweis: Ein Schachbrett besteht aus 8×8 quadratischen Feldern, die in acht Zeilen zu je acht Feldern angeordnet sind. Eine Dame, die auf Feld F steht, bedroht alle Felder in derselben Zeile, in derselben Spalte und auf den beiden Diagonalen, die durch F gehen.)
Wie berechnet man Levenshtein-Distanz?
Geben Sie ein oder zwei Beispiele an, das man mit Backtracking gut lösen kann
Wie kann man das Rucksack-Problem mit Brute-Force lösen? Laufzeit?
Wie funktioniert der Algorithmus Best-first-Search?
Was ist Pruning (Synonyme: Abschneiden, Cutt off)
Was ist der A*-Algorithmus und wie funktioniert er?
Was ist die Levenshtein-Distanz?
Welche Operationen werden beachtet?
Welche Entwurfsmuster für Algorithmen kennen Sie?
Findet ein Greedy-Algorithmus immer eine optimale Lösung?
Wenn nein, kennen Sie ein Beispiel?
Geben Sie ein Beispiel für einen Greedy-Algorithmus aus der Vorlesung
Warum kann man nicht jedes Problem mit Brute-Force lösen?
Findet Backtracking immer eine optimale Lösung?
Erklären Sie kurz, was der Minmax-Algorithmus ist. Nennen Sie ein Beispiel, in welchem ein Minmax-Algorithmus eingesetzt wird.
Aus der letzten Lektion: Wie funktionieren Las-Vegas-Algorithmen? Nennen Sie ein Beispiel für einen derartigen Algorithmus.
Aus der letzten Lektion: Wie funktionieren Monte-Carlo-Algorithmen? Nennen Sie ein Beispiel für einen derartigen Algorithmus.
Aus der letzten Lektion: Wie funktionieren Genetische Algorithmen? Nennen Sie ein Beispiel für einen derartigen Algorithmus.
Aus der letzten Lektion: Wie funktionieren Simulated Annealing? Nennen Sie ein Beispiel für einen derartigen Algorithmus.
Aus der letzten Lektion: Wie funktionieren dynamische Programmierung? Nennen Sie ein Beispiel für einen derartigen Algorithmus.

Hashing

Erklären Sie, was eine Hash-Table ist, und welches deren grundlegende Merkmale und Operationen sind.

Hashing: Was versteht man unter einer Kollision und warum muss es (in der Regel) Kollisionen geben?

Hashing: Erklären Sie, warum trotz Kollisionen beim Hashing die Elemente wiedergefunden werden.

Welches sind die Hauptunterschiede zwischen Hash-Tabellen und Listen, wenn ein konkretes Element in der Datenstruktur gesucht wird?

Erläutern Sie Unterschiede und Gemeinsamkeiten zwischen AVL-Trees und Hash-Tables.

Hashing: Skizzieren Sie quadratisches Probing. Wie geht man vor?

Sie machen Hashing mit linearem oder quadratischem Sondieren. Sie rechnen damit, dass in Ihrer Hashtabelle mehrfach eingefügt, gelöscht und gesucht wird. Die Reihenfolge, in der das geschieht, kennen Sie nicht. Um den Eintrag in der Tabelle "map" am Index h zu löschen, schreiben Sie "map[h] = null". Warum ist das falsch und was müssen Sie stattdessen tun?

Sie machen Hashing mit linearem oder quadratischem Sondieren.

Sollten Sie den Füllgrad der Tabelle beim Löschen eines Elements nach unten korrigieren oder nicht? Warum?

Sie machen Hashing mit linearem oder quadratischem Sondieren. Sie haben in Ihrer Hashtabelle schon mehrfach Elemente eingefügt, gesucht und gelöscht. Nun treffen Sie beim Suchen eines Elements auf einen Eintrag in der Tabelle, der null ist. Was schließen Sie hieraus?

Erklären Sie wie Extendible Hashing funktioniert.

Suchen

Sie möchten in einem Text mit dem Knuth-Morris-Pratt-Algorithmus nach dem Muster ABCDEFGH suchen. Wie sieht die next-Tabelle aus? Begründen Sie.

Sie möchten in einem Text mit dem Knuth-Morris-Pratt-Algorithmus nach dem Muster BANANANANANANA suchen. Wie sieht die next-Tabelle aus? Begründen Sie.

Gegeben sei eine unsortierte verkettete Liste von int-Werten. Gegeben seien die beiden Methoden search1() und search2(). Beide Methoden suchen eine Zahl x in der Liste l und geben eine Referenz auf das erste gefundene Listenelement zurück, wenn x gefunden wurde und null sonst.

(Wenn Ihnen die Bedeutung der Methoden getFirst(), getNext(), getValue(), appendAtEnd() oder remove() nicht klar ist, dürfen Sie fragen.)

Nehmen Sie an, dass die Operationen getFirst(), getNext(), getValue(), appendAtEnd() und remove() in $O(1)$ Zeit ablaufen. Nehmen Sie weiter an, dass Nebenläufigkeit für die Korrektheit der Methoden keine Rolle spielt.

Angenommen, Sie haben es in Ihrer Anwendung mit besonders langen Listen zu tun, in denen sie häufig nach Werten suchen müssen. Welche der beiden Methoden verwenden Sie? Begründen Sie.

Was ist ein invertierter Index?

Wie funktioniert die Trigramm-Suche?

Wie funktioniert die phonetische Suche (Soundex)?

Java

Erklären Sie die Interfaces Comparable und Comparator.

Erklären Sie kurz die Collections-Klasse von Java, sowie ein zwei Methoden der Klasse.

Bei welchen Konstrukten können generische Typen (Typparameter) verwendet werden und an welcher Stelle im Code muss der Typparameter aufgeführt werden.

Können generische Methoden in nicht generischen Klassen verwendet werden?

Erklären Sie kurz das Visitor-Pattern.

Nennen Sie zwei Beispiele von Java-Klassen, welche Hashing verwenden.

Erklären Sie wie Regex funktioniert und geben Sie drei Beispiele für Metasymbole und wie diese funktionieren.

Wofür wird die Collator-Klasse in Java verwendet?

Welcher Sortieralgorithmus wird in der Java-Klasse Arrays angewendet, wenn ein Array von primitiven Typen sortiert werden soll?

Welcher Sortieralgorithmus wird in der Java-Klasse Arrays angewendet, wenn ein Array von Objekten sortiert werden soll?
Was ist das Java-Memory-Modell und wie sieht es aus?
Welche Probleme können entstehen, weil das Java-Memory-Modell und das Hardware-Memory-Modell nicht identisch sind?
Erklären Sie, was in Java ein Thread-Pool ist. Was ist der Unterschied zwischen einem Thread und einem Task?
Was wird mit dem Future-Interface bezweckt?
Wofür ist der ForkJoinPool von Java gedacht?
Nennen Sie Algorithmen-Beispiele für eine automatische Speicherverwaltung (Garbage Collection).
Beschreiben Sie wie der Mark-Compact-Algorithmus (Garbage Collection) funktioniert.
Beschreiben Sie wie der Copying-GC-Algorithmus (Garbage Collection) funktioniert.
Was ist die Idee bei Generational-GC (Garbage Collection)?

Code und Texte

Code 1	<pre>sum = 0; i = 1 while i < N i = i*2 sum += log(i) print(sum)</pre>
Code 2	<pre>sum = 0; for i = 1 .. n for j = 1 to n sum += i*j print(sum)</pre>
Code 3	<pre>sum = 0; for i = 1 .. N for j = 1 to i for k = 1 to j sum += i*j print(sum)</pre>
Code 4	<pre>sum = 0; for i = 1 .. N for j = 1 to M for k = 1 to M sum += i*j print(sum)</pre>
Code 5	<p>Gegeben ein Array F der Länge n</p> <pre>for i = 1 to n x = binarySearch(i in F)</pre>
Code 6	<pre>public int foo(int m, int n) { assert(m >= 0); assert(n >= 0); if (m == 0) { return n + 1; } else if (n == 0) { return foo(m - 1, 1); } else return foo(m - 1, foo(m, n - 1)); }</pre>

Code 7	<pre> public int foo(int n) { assert(n >= 0); if (n == 0) { return 1; } else { return n + foo(n - 1); } } </pre>
Code 8	<pre> public int wrongFac(int n) { assert(n >= 0); return n * wrongFac(n-1); } </pre>
Code 9	<pre> public ListItem search1(IntegerList l, int x) { ListItem p = l.getFirst(); while (p != null && p.getValue() != x) { p = p.getNext(); } return p; } public ListItem search2(IntegerList l, int x) { ListItem s = l.appendAtEnd(x); ListItem p = l.getFirst(); while (p.getValue() != x) { p = p.getNext(); } ListItem ret = p == s ? null : p; l.remove(s); return ret; } </pre>

Text 1 10, 1, 9, 2, 8, 3, 7, 4

Bilder

Bild 1

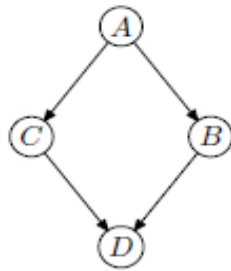


Abbildung 1: Vielleicht ein Baum

Bild 2

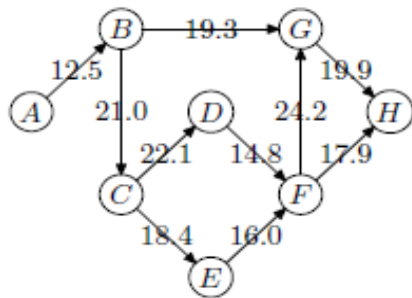


Abbildung 3: Ein Fluss.

Bild 3

