

Fragen	Pseudo-Code / Bild
Schreiben Sie einen Algorithmus in Pseudo-Code, der überprüft, ob eine Zahl eine Primzahl ist. Tipp: Verwenden Sie die Modulo-Funktion.	
Was ist der Unterschied zwischen einer Queue und einem Stack?	FIFO und LIFO
Was ist ein Stack; welches sind die grundlegenden Operationen; wie kann er implementiert werden?	PUSH, POP, PEAK
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 $x \geq 0$. Nennen Sie eine möglichst genaue Nachbedingung.	
Was ist eine Queue; welches sind die grundlegenden Operationen; wie kann sie implementiert werden; was ist eine PriorityQueue und wie funktioniert sie?	
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.	
Wie ist die Worst-Case-Laufzeit von binärer Suche? Wieso?	$\log(n)$. komplette Hälfte durchsuchen
Erklären Sie wie binäre Suche funktioniert.	Array halbieren
Welche Laufzeit hat folgender Code - grobe Abschätzung genügt? <div>$\log(n)$</div>	<pre>sum = 0; i = 1 while i < N i = i*2 sum += log(i) print(sum)</pre>
Nennen Sie einige typische Komplexitätsklassen	$O(1)$, $O(n)$, $O(n^2)$
1. Was hält die Omega-Notation fest?	Untere Komplexitätsgrenze (mindestens)
1. Was hält die Theta-Notation (Big- Θ) fest und 2. wie lautet die Definition der Theta-Notation?	Genaue Komplexität (genau) $\Theta(g(n)) = f(n)$ positive c_1, c_2 and n_1 $0 \leq c_1(g(n)) \leq f(n) \leq c_2(g(n))$ for all $n \geq n_1$
Was ist der Unterschied zwischen Average und Worst Case Laufzeit?	Worst, kann nicht höher sein
Falls immer $f(x) < g(x)$ ist, ist dann $f = O(g)$?	
Hat folgender Algorithmus Laufzeit $O(n^3)$? <div>nein, n^2</div>	<pre>sum = 0; for i = 1 .. n for j = 1 to n sum += i*j print(sum)</pre>
1. Was hält die O-Notation (Big-O) fest und 2. wie lautet die Definition der O-Notation? <div>"Worst"-Case Laufzeit</div>	Wenn $T(n) = O(g(n))$, gilt für n_0 die Gleichung $T(n) \leq c * g(n)$
Welche Laufzeit hat die Suche in einer unsortierten LinkedList? Was ist, wenn die Liste sortiert ist?	$O(n)$ und $O(n)$
Welche Laufzeit hat folgender Code? <div>n^3</div>	<pre>sum = 0; for i = 1 .. N for j = 1 to i for k = 1 to j sum += i*j print(sum)</pre>
Welche Rechenregeln gelten bei der Landau-Notation? (z.B. $O(n^2 + n^3) \rightarrow O(n^3)$)	c-Faktoren weg und dann grösster n-Wert zählt
Welche Laufzeit hat folgender Code? <div>$O(n * m^2) = O(n^3)$</div>	<pre>sum = 0; for i = 1 .. N for j = 1 to M for k = 1 to M sum += i*j print(sum)</pre>
Nennen Sie für vier Komplexitätsklassen (O-Notation) einen Algorithmus mit dieser Laufzeit.	
Welche Laufzeit hat folgender Code? <div>$O(n * \log(n))$</div>	Gegeben ein Array F der Länge n for i = 1 to n x = binarySearch(i in F)
Warum sind iterative Algorithmen oft schneller als ihre rekursiven Varianten?	kein Laufzeit Overhead
Wie lautet die Definition der Fibonacci-Zahlen?	
Wie kann man die Fibobacci-Zahlen iterativ berechnen?	
Wie lautet die Rekursionsgleichung für die maximale Anzahl Vergleiche bei binärer Suche	
Nennen Sie ein Beispiel rekursiver Programmierung aus der Vorlesung.	
Wie kann man die Fibonacci-Zahlen rekursiv berechnen? Was passiert dabei mit der Laufzeit <div>$O(2^n)$</div>	
Was ist eine Endrekursion?	Rekursion am Ende von else()

```
int fak(int n){
    if (n == 0) return 1;
    else return n* fak(n-1);
}
```

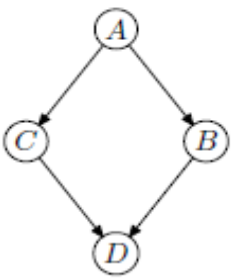
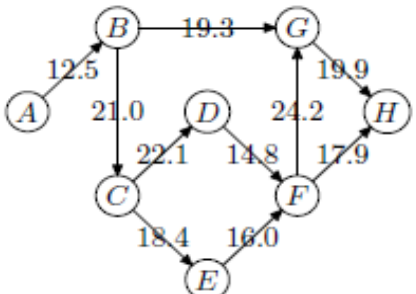
$$fib(n) = \begin{cases} 0 & \text{falls } n = 0 \\ 1 & \text{falls } n = 1 \\ fib(n-1) + fib(n-2) & \text{sonst} \end{cases}$$

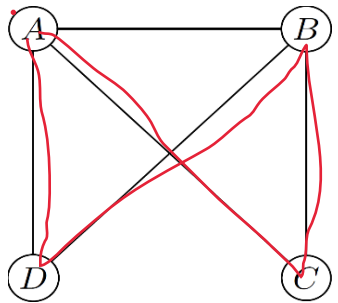
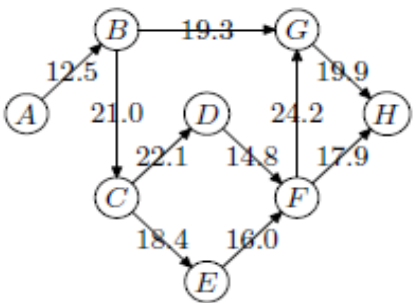
n	0	1	2	3	4	5	6	7	8	9	10	11	12	..
fn	0	1	1	2	3	5	8	13	21	34	55	89	144	..

```
public int fib(int n) {
    if (n == 0) return 0;
    else if (n == 1) return 1;
    else return fib(n-1) + fib(n-2);
}
```

```
int fib(int n) {
    int res = 0, zwRes0 = 1, zwRes1 = 1;
    if (n == 0) return 0;
    if (n == 1) return 1;
    for (int i = 2; i <= n; i++) {
        res = zwRes0 + zwRes1;
        zwRes0 = zwRes1;
        zwRes1 = res;
    }
    return res;
}
```

Ist folgender Code endrekursiv?	<div>Ja</div> <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>
Ist folgender Code endrekursiv?	<div>ja</div> <pre> public int foo(int n) { assert(n >= 0); if (n == 0) { return 1; } else { return n + foo(n - 1); } } </pre>
<p>Aus der Denition der Fakultät $n! = 1 \times 2 \times \dots \times n$ kann man die Beziehung $n! = n(n - 1)!$ herleiten.</p> <p>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?</p>	<pre> public int wrongFac(int n) { assert(n >= 0); return n * wrongFac(n-1); } </pre> <div>bei $n = 0$ muss 1 genommen werden, oder $\text{assert}(n > 0)$, sonst $0 * 1 * n$</div>
Definieren Sie den Datentyp "Liste" rekursiv.	Listnode<T> next;
Skizzieren Sie eine rekursive Methode, welche die Elemente einer einfach verketteten Liste in umgekehrter Reihenfolge ausgibt (Pseudocode).	<div> if(node.next != null -> next() sout(node.get()) </div>
Was ist der Unterschied zwischen einer einfach und einer doppelt verketteten Liste?	next, prev. vor und zurück navigieren
Wie kann man in einer einfach verketteten Liste ein Element an Position i einfügen?	durchnavigieren und Position zählen
Welche Laufzeit hat das Einfügen eines Elements an der ersten Position in einer einfach-verketteten Liste?	O(1)
Nennen Sie zwei Implementierungen von Listen in Java	ArrayList, LinkedList
Welche Methoden hat ein Java Iterator?	next, hasNext, remove
Wie verwendet man einen Java Iterator?	Array.iterator() dann mit schleife
Was ist in Java der Unterschied zwischen einer LinkedList und einer ArrayList? Was bedeutet das für die Performance.	
Welche Laufzeit hat das Löschen eines Elementes in einer doppelt verketteten Liste?	O(n)
Welche Laufzeit hat das Einfügen eines Elements an der ersten Position in einer ArrayList?	O(1)
Warum ist ein "tail"-pointer bei Listen oft hilfreich?	muss nicht durch die ganze Liste gehen fürs Ende
<p>Gegeben sei ein unbekannter ADT mit zwei Operationen void foo(int x) int bar(). Sie rufen nun foo() und bar() nacheinander in folgender Reihenfolge auf:</p> <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> <p>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.</p>	<div> stack put(1) put(2) pop() put(3) put(4) pop() pop() pop() </div>
Nennen Sie eine Operation, die typischerweise in Java in einer LinkedList schneller geht als in einer ArrayList. Wieso?	Mit Iterator add/insert
Gegeben ist eine einfach-verkettete Liste OHNE Tail-Pointer. Welche Datenstruktur sollte man damit NICHT implementieren, einen Stack oder eine Queue?	Stack
Nennen Sie eine Operation, die typischerweise in Java in einer ArrayList schneller ist als in einer LinkedList	Zugriff auf Element i array.get(i) = O(1)

Ist das ein Baum (Abbildung 1)? Begründen Sie!	<div> nein. nur ein gerichteter graph. Node können nicht auf gleichen leaf zeigen Baum = nur ein Weg pro Node </div>  <p>Abbildung 1: Vielleicht ein Baum</p>
Was ist ein binaerer Suchbaum?	links < Root <= rechts
Was ist ein binärer Baum?	max 2 children
Welche Traversierungsarten bei Bäumen kennen Sie?	Pre,Post, level, In- Order
Worin unterscheiden sich Implementierungen von pre- in- und postorder?	root, left, right left, root, right left, right, root
Was passiert, wenn Sie die Knoten eines binaeren Suchbaums inorder ausgeben?	sortierte Ausgabe
Gegeben sei ein Baum, bei dem die inneren Knoten Operationen sind wie + - * / und die Blaetter Zahlen. Was erhalten Sie bei Inorder,Preorder Postorder?	
Fuegen Sie die Zahlen der Reihe nach in einen binaeren Suchbaum ein. Was erkennen Sie?	10, 1, 9, 2, 8, 3, 7, 4 "linksseitig"
In einem bin. Suchbaum mit ungerade vielen Elementen ($n = 2k+1$) steht an jedem Knoten, wieviele Knoten der gesamte Teilbaum hat. Wie kann man mit dieser Information den Median bestimmen?	root = Total Anzahl? dann check links/rechts = median
Wo findet man in einem binären Suchbaum den grössten Knoten?	ganz rechts aussen
Wie hoch ist ein binaerer Baum mit n Elementen mindestens und hoechstens?	min: $\log(n) / \log(2) = x \Rightarrow 2^x = n$ max : n
Skizzieren Sie einen vollständig gefüllten, binären Suchbaum der Höhe 3 und löschen Sie anschliessend den gezeigten Knoten aus dem Baum.	
Kann man aus der Pre- und Postorder-Reihenfolge einen Baum immer eindeutig wiederherstellen?	ja
Was macht man, wenn ein Knoten in einem B-Baum "überläuft"	Knoten aufteilen
Skizzieren Sie das Problem bei binaeren Suchbaeumen, das zu balancierten Baeumen fuehrt	Baum kann sortiert sein, aber nicht balanciert. Alles rechts oder alles links => somit $O(n)$
Wie lautet die Definition eines AVL Baums?	Höhe links - Höhe rechts <= 1
Welche Kompletität hat eine Rotation in einem AVL-Baum?	$O(1)$
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?	Einfach nur wenn X und > gleich gross
Welche Operationen werden benoetigt, um einen durch einfügen unbalancierten AVL-Baum wieder zu balancieren? Beschreiben Sie diese anhand einer Skizze	
Welche Komplexitaet hat das Einfügen in einen AVL-Baum?	$\log(n)$
Warum hat das Einfügen/Suchen in AVL-Bäumen logarithmische Laufzeit?	da sortiert und ca nur die Hälfte durchsucht werden muss
Was ist ein B-Baum der Ordnung k?	K = max Anzahl kinder pro Knoten ausser Root
Wieviele Knoten (Pages) hat ein B-Baum der Ordnung k und Höhe h hoechstens?	
Wofür werden B-Bäume in der Praxis eingesetzt	Datenbankindex, Dateisysteme
Wie funktioniert Einfügen in einem B-Baum?	
Wie wird die Höhe eines B-Baumes erhöht?	
Was ist Breitensuche? Was ist Tiefensuche?	Tief: start -> tief -> Sackgasse -> zurück bis unbesucht -> tiefer (Stack) (Queue) Breit: Start -> nachbar -> tiefer -> nachbar
Was ist eine Adjazenzmatrix?	"Verbunds"-matrix. NxN-Matrix zeigt Verbindung zu den jeweiligen Knoten
Betrachten Sie den gerichteten gewichteten Graphen in Abbildung 3. Bestimmen Sie den maximalen Fluss von A nach H.	<div>A-B-G-H. weil A-B 12.5 ist jeder Fluss maximal</div>  <p>Abbildung 3: Ein Fluss.</p>
Wieviele Kanten hat ein vollständiger (kompletter) ungerichteter Graph mit n Knoten?	$n(n-1) / 2$
Was ist ein minimaler Spannbaum?	Graph in Graph, mit geringstem Gewicht
Wie lautet die Definition eines ungerichteten (gerichteten, gewichteten) Graphen?	Pfeile ja/nein, Kosten pro Linie
Wie kann man einen ungerichteten Graphen als gerichteten Graphen repräsentieren?	Pfeile in beide Richtungen
Was ist ein Spannbaum?	Graph in Graph, welcher jeden Knoten enthält
Sind minimale Spannbäume eindeutig bestimmt? Begründung/Gegenbeispiel	

Angenommen es gibt in einem Graphen genau eine Kante minimalen Gewichts. Muss die in einem minimalen Spannbaum vorhanden sein? Begründung	ja. Kante gehört zu einem Konten, welcher besucht werden muss -> somit immer drin
Gegeben ist ein Spannbaum eines ungerichteten Graphen. Nun verbinden Sie zwei bisher nicht verbundene Knoten. Ist das Resultat immer noch ein Spannbaum?	Ja
Welche Graphenrepräsentation würden Sie verwenden, um einen Graphen als Textfile zu speichern?	ungerichtet, da egal in welche richtung das "pixel" gezeichnet wird
Sind Spannbäume eindeutig bestimmt? Wenn ja, begründen Sie, wenn nein geben Sie ein Gegenbeispiel.	.
Sie haben einen unger. Graphen mit den 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 berechnet der Algorithmus von Dijkstra? Wie funktioniert er? Wieso ist das ein Beispiel fuer einen Greedy-Algorithmus?	Nutzt das Gewicht der Kanten um den besten Pfad zu finden. (Checkt aber leider alle Pfade)
Gegeben sei der gerichtete Graph aus Abbildung 3. Wie lautet die Reihenfolge der Knoten beim Durchlaufen des Graphen in Breitensuche, wenn man beim Knoten A beginnt?	<div>ABCGEDHF</div>  Abbildung 3: Ein Fluss.
Wie funktioniert Insertionsort?	2 Teile: sortiert;unsortiert. Element wählen und an richtige Stelle tauschen (jedes Element nach rechts VERSCHIEBEN)
Wie funktioniert Quicksort?	An Pivot-Element aufteilen und sortieren. left < Pivot < Right -> bis Partition nur noch aus 1/2 Elementen besteht (Rekuriv)
Warum ist die Worse-Case-Laufzeit von Quicksort quadratisch?	Wenn sortiert/gleich, dann kann nicht geteilt werden (log(n)) -> somit n^2 durchläufe
Wie funktioniert Selectionsort?	2 Teile: sortiert;unsortiert. Kleinstes Element an Richtige stelle VERTAUSCHEN
Wie ist die Laufzeit von BubbleSort bei absteigend oder aufsteigend sortierter Folge?	immer O(n^2)
Wie funktioniert Bubblesort? Was ist die Laufzeit, und warum?	immer O(n^2). von l-r, grösseres Element tauschen. 1. Lauf = grösstes element rechts
Definieren Sie einen Sortieralgorithmus	
Welche Eigenschaften hat SelectionSort? - stabil (Sortierung bleibt erhalten)? - in situ (zusätzliche Speicherbedarf ist unabhängig von der Anzahl der zu sortierenden Elemente)? - Laufzeit?	instabil, O(n^2)
Welchen Sortieralgorithmus würden Sie nehmen, wenn Sie die Einträge in einer Datei sortieren wollen, die nicht komplett in den Hauptspeicher passt?	merge-sort
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?	sortiert weil die möglichen Werte bereits bekannt sind (Postfach) funktion für Schlüssel
Wie beeinflusst bei Quicksort die Wahl des Pivotelements das Laufzeitverhalten, wenn das Array schon sortiert ist?	richtiges Pivot wählen, damit beide Teile gleich gross sind
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?	Hashing geht nicht, weil aus dem Hash-Wert nicht auf den Inhalt geschlossen werden kann. Der Hash-Wert sagt nichts über die Grösse der Elemente aus (schlecht bei Sortierung).
Eine Folge von Zahlen die aufsteigend sortiert werden soll, ist bereits absteigend sortiert. Wie viele Vergleiche und Vertauschungen brauchen Bubblesort und Selection Sort?	immer noch n^2
Was ist die Levenshtein-Distanz?	anzahl-operationen bis wort veränderung. Update, delete, insert
Welche Operationen werden beachtet?	
Geben Sie ein Beispiel an, das man mit Backtracking gut lösen kann	labyrinth
Welche Entwurfsmuster für Algorithmen kennen Sie?	
Wie kann man das Rucksack-Problem mit Brute-Force lösen? Laufzeit?	könnte man, dauert aber zu lange. O(2^n)
Wie berechnet man Levenshtein-Distanz?	zusammenrechnen der operationen

Findet ein Greedy-Algorithmus immer eine optimale Lösung? Wenn nein, kennen Sie ein Beispiel?	nein. kann in ein lokales minima/maxima steckenbleiben
Was ist der Unterschied zwischen Lokaler Suche und Random Sampling?	
Geben Sie ein Beispiel für einen Greedy-Algorithmus aus der Vorlesung	Dijkstra, oder min Spannbaum
Warum kann man nicht jedes Problem mit Brute-Force lösen?	jedes Kombination testen, wäre zu viel Aufwand/zeit
Findet Backtracking immer eine optimale Lösung?	nein, aber en mögliche
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 x 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.)	
Hashing: Was versteht man unter einer Kollision und warum muss es (in der Regel) Kollisionen geben?	unterschiedlicher Key = gleicher Index (hash)
Hashing: Erklären Sie, warum trotz Kollisionen beim Hashing die Elemente wiedergefunden werden.	durch lineares / quadratisches Sondieren
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?	map[h] muss nicht gleich key sein
Hashing: Skizzieren Sie quadratisches Probing. Wie geht man vor?	
Sie machen Hashing mit linearem oder quadratischen Sondieren. Sollten Sie den Füllgrad der Tabelle beim Löschen eines Elements nach unten korrigieren oder nicht? Warum?	downsizing ist zu viel aufwand => alles neu hashen. bei vielen löschen/insert wird das zu oft gemacht, also nicht nötig
Sie machen Hashing mit linearem oder quadratischen 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?	element ist nicht vorhanden oder HashTable ist falsch implementiert
Sie möchten in einem Text mit dem Knuth--Morris--Pratt-Algorithmus nach dem Muster ABCDEFGH suchen. Wie sieht die next-Tabelle aus? Begründen Sie.	alles geht auf 0, da kein pattern
Sie möchten in einem Text mit dem Knuth-Morris-Pratt-Algorithmus nach dem Muster BANANANANANANA suchen. Wie sieht die next-Tabelle aus? Begründen Sie.	Das wiederholende Pattern "AN" wird verknüpft. Beim letzten A kann einfach aufs vorletzte verwiesen werden
Gegeben sei eine unsortierte verkettete Liste von int-Werten. Gegeben seien die beiden Methoden search1() und search2(). (Wenn Ihnen die Bedeutung der Methoden getFirst(), getNext(), getValue(), appendAtEnd() oder remove() nicht klar ist, dürfen Sie fragen.) 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. 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.	<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>
	search1 weil ich nicht die original-liste verändern will. Ist evtl gar nicht möglich