

<p>S. 11 Mitte</p> <p>31.10.11</p>	<p><b>FALSCH:</b> wie wir sehen werden, gibt es zusätzliche Schlüsselwörter in C++ wie z. B. <code>class</code>, <code>public</code> oder <code>private</code>.</p> <p><b>RICHTIG:</b> wie wir sehen werden, gibt es zusätzliche Schlüsselwörter in C++ wie z. B. <code>class</code>, <code>public</code> oder <code>private</code>.</p>
<p>S. 28 oben S. 29 oben</p> <p>2.1.2016</p>	<p><b>STATT:</b> <code>vector&lt;T&gt;::size_type i;</code></p> <p><b>BESSER:</b> <code>typename vector&lt;T&gt;::size_type i;</code></p> <p><b>BEMERKUNG:</b> Wird von manchen Compilern zwar übersetzt, ist aber erst durch das Schlüsselwort <code>typename</code> standardkonform.</p>
<p>S.32 Mitte</p> <p>2.1.2016</p>	<p><b>FALSCH:</b> Wenn die Variable <code>v</code> vom Datentyp <code>vector[double]</code> ist, ...</p> <p><b>RICHTIG:</b> Wenn die Variable <code>v</code> vom Datentyp <code>vector&lt;double&gt;</code> ist, ...</p>
<p>S. 46 oben</p> <p>20.4.11</p>	<p><b>FALSCH:</b></p> <pre> struct BKnoten {     struct Knoten *left;    // Zeiger auf linken Nachfolger     struct Knoten *right;  // Zeiger auf rechten Nachfolger     int id;                // Benutzer-ID     short priority;     char name[STR_SIZE]; }; </pre> <p><b>RICHTIG:</b></p> <pre> struct BKnoten {     struct BKnoten *left;  // Zeiger auf linken Nachfolger     struct BKnoten *right; // Zeiger auf rechten Nachfolger     int id;                // Benutzer-ID     short priority;     char name[STR_SIZE]; }; </pre>
<p>S. 49 Mitte oben</p> <p>2.1.2016</p>	<p><b>FALSCH:</b> <b>Induktionsschritt:</b> Es sei nun <math>k_0 \geq 0</math>.</p> <p><b>RICHTIG:</b> <b>Induktionsschritt:</b> Es sei nun <math>k_0 \geq 1</math>.</p>
<p>S. 57 Mitte</p> <p>31.10.11</p>	<p><b>FALSCH:</b> hat die Wurzel den Wert <math>w</math> und gilt <math>w &lt; s</math>, dann durchsuche den linken Teilbaum der Wurzel;</p> <p><b>RICHTIG:</b> hat die Wurzel den Wert <math>w</math> und gilt <math>s &lt; w</math>, dann durchsuche den linken Teilbaum der Wurzel;</p>

S. 95 oben 31.10.11	<b>FALSCH:</b> #include "Stack2 .hpp"	<b>RICHTIG:</b> #include "stack2.hpp"
S. 104 oben 31.10.11	<b>FALSCH:</b> #include "Stack1a .hpp"	<b>RICHTIG:</b> #include "stack1a.hpp"
S. 105 oben 31.10.11	<b>FALSCH:</b> #include "Stack3 .hpp"	<b>RICHTIG:</b> #include "stack3.hpp"
S. 105 oben  Kommentar zum Quelltext Stack3::pop()  23.11.10	<b>FALSCH:</b> // Ist der Stack leer, dann wird -1 zurückgeben  <b>RICHTIG:</b> // Ist der Stack leer, dann wird fehlerwert zurückgeben	
S. 129 oben    23.11.10	<b>FALSCH:</b>  <b>Satz 4.1</b> Existiert der Grenzwert $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)}$ (und ist endlich), dann folgt $g(n) = O(f(n))$  <b>RICHTIG:</b>  <b>Satz 4.1</b> Existiert der Grenzwert $\lim_{n \rightarrow \infty} \frac{g(n)}{f(n)}$ (und ist endlich), dann folgt $g(n) = O(f(n))$	
S. 129 oben  Bemerkung:   23.11.10	Ebenso müssen im Beweis die Funktionen $f$ und $g$ vertauscht werden: <b>FALSCH:</b> Aus $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = D < \infty$ folgt sofort aus der Definition des Grenzwerts, dass es eine natürliche Zahl $n_0$ gibt, sodass $f(n) \leq (D+1) \cdot g(n)$ für alle $n \geq n_0$ . Also gilt $g(n) = O(f(n))$ . <b>RICHTIG:</b> Aus $\lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = D < \infty$ folgt sofort aus der Definition des Grenzwerts, dass es eine natürliche Zahl $n_0$ gibt, sodass $g(n) \leq (D+1) \cdot f(n)$ für alle $n \geq n_0$ . Also gilt $g(n) = O(f(n))$ .	
S. 138 unten    23.11.10	<b>FALSCH:</b> Im $i$ -ten Aufruf wird $V(n)$ um den Summanden $i - 1$ erhöht; insgesamt erhalten wir somit ein quadratisches Worst-Case-Verhalten: $V(n) = \sum_{i=2}^n (i-1) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} = \Theta(n^2)$ <b>RICHTIG:</b> Im $i$ -ten Aufruf wird $V(n)$ um den Summanden $n - i$ erhöht; insgesamt erhalten wir somit ein quadratisches Worst-Case-Verhalten: $V(n) = \sum_{i=1}^n (n-i) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} = \Theta(n^2)$	

<p>S. 146 unten</p> <p>2.1.2016</p>	<p><b>FALSCH:</b></p> $E := \sum_{i=1}^b t_i$ <p><b>RICHTIG:</b></p> $E := \frac{1}{b} \sum_{i=1}^b t_i$
<p>S. 177 unten</p> <p>2.1.2016</p>	<p><b>STATT:</b></p> <p>Auf dasselbe Ergebnis kommt man auch, wenn zuerst Spalten- und danach Zeilenreduktionen durchgeführt werden.</p> <p><b>BESSER:</b></p> <p>Auf dasselbe Ergebnis kommt man auch in diesem Beispiel, wenn zuerst Spalten- und danach Zeilenreduktionen durchgeführt werden. Im Allgemeinen können sich aber unterschiedliche Schranken ergeben.</p>
<p>S. 182 oben</p> <p>20.4.11</p>	<p><b>FALSCH:</b></p> <p>Die ersten Elemente der Fibonacci-Folge <math>\{f(n)\}_{n \in \mathbb{N}}</math> lauten also 0, 1, 2, 3, 5, 8, 13, 21, 34, ...</p> <p><b>RICHTIG:</b></p> <p>Die ersten Elemente der Fibonacci-Folge <math>\{f(n)\}_{n \in \mathbb{N}}</math> lauten also 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...</p>
<p>S. 182 unten</p> <p>20.4.11</p>	<p><b>FALSCH:</b></p> <p>Danach wird mithilfe der Tabellenwerte der Wert <math>f(3)</math> berechnet und das Ergebnis in der Tabelle gespeichert. Ebenso wird mit den Werten <math>f(4), f(5), \dots, f(9)</math> verfahren ...</p> <p><b>RICHTIG:</b></p> <p>Danach wird mithilfe der Tabellenwerte der Wert <math>f(2)</math> berechnet und das Ergebnis in der Tabelle gespeichert. Ebenso wird mit den Werten <math>f(3), f(4), \dots, f(9)</math> verfahren ...</p>
<p>S. 211 Mitte</p> <p>20.4.11</p>	<p><b>FALSCH:</b></p> <p>Die Postorder-Ausgabe liefert folgende Werte:</p> <p style="text-align: center;">50, 20, 10, 32, 35, 30, 64, 70, 60, 90, 80, 40</p> <p><b>RICHTIG:</b></p> <p>Die Postorder-Ausgabe liefert folgende Werte:</p> <p style="text-align: center;">50, 20, 10, 32, 35, 30, 64, 70, 60, 90, 80, 14</p>
<p>S. 226 Mitte</p> <p>31.10.11</p>	<p><b>FALSCH:</b></p> <p><code>#include "Stack3 .hpp"</code></p> <p><b>RICHTIG:</b></p> <p><code>#include "stack3.hpp"</code></p>
<p>S. 227 oben</p> <p>31.10.11</p>	<p><b>FALSCH:</b></p> <p><code>#include "Stack4 .hpp"</code></p> <p><b>RICHTIG:</b></p> <p><code>#include "stack4.hpp"</code></p>

<p>S. 233 <i>Mitte</i></p> <p>2.1.2016</p>	<p><b>FALSCH:</b> Insgesamt ergibt sich die Schranke <math>19 + 0 = 19</math>. <del>Allgemein kann man zeigen, dass sich stets dasselbe Ergebnis ergibt, unabhängig davon, ob zuerst die Spalten- oder die Zeilenreduktionen durchgeführt werden.</del></p> <p><b>RICHTIG:</b> Insgesamt ergibt sich die Schranke <math>19 + 0 = 19</math>.</p> <p><b>BEMERKUNG:</b> Wie das Beispiel der nächsten Übungsaufgabe zeigt, können sich durchaus unterschiedliche Schranken ergeben, wenn die Reihenfolge der Zeilen- oder die Spaltenreduktionen vertauscht wird.</p>
--	--