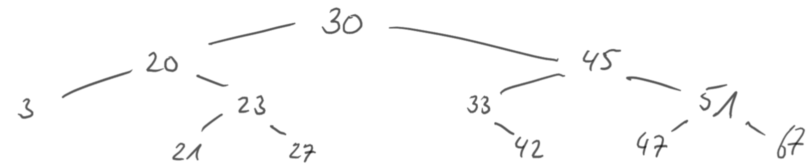
Algorithmen und Datenstrukturen
Übungsserie 6

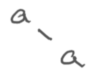
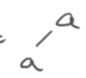
Markus Pawellek – 144645 Übung: Mo 10 – 12

Aufgabe 1

(a) zugehöriger BST :



Seien $n \in \mathbb{N}$ und $N_n := \{k \in \mathbb{N} \mid k \leq n\}$. Weiterhin seien X eine Menge mit Ordnungsrelation, B_n die Menge der binären Suchbäume auf X mit n Knoten und $M_n := \{f : N_n \to X\}$ die Menge aller n-elementigen Sequenzen auf X. Die Abbildung $p : B_n \to M_n$ soll gerade jedem Baum $b \in B_n$ die Postorderreihenfolge $x \in M_n$ seiner Schlüssel zuordnen. Dann ist ein Baum $b \in B_n$ gerade eindeutig durch seine Postorderreihenfolge $p(b)$ gegeben, wenn p injektiv ist.

Sei $a \in X$. $b_1 :=$  $b_2 :=$ 

$\Rightarrow$ $b_1, b_2 \in B_2$, $b_1 \neq b_2$, $p(b_1) = (a,a) = p(b_2)$

$\Rightarrow$ p ist auf B_n nicht injektiv $\Rightarrow$ beliebige binäre Suchbäume sind nicht durch Postorderreihenfolge eindeutig bestimmt.

Sei $\tilde{B}_n \subset B_n$, sodass alle $b \in \tilde{B}_n$ gerade n paarweise verschiedene Schlüssel enthalten. Dann sei $\tilde{p} := p|_{\tilde{B}_n}$ die Einschränkung von p auf $\tilde{B}_n$.

Auch hier gilt wieder, dass alle $b \in \tilde{B}_n$ eindeutig durch $p(b)$ bestimmt sind, wenn $\tilde{p}$ injektiv ist.

Konstruktion: Sei $x \in M_n$ gegeben durch $x = p(b)$ für ein $b \in \tilde{B}_n$.

(1) Die Wurzel von b ist durch x_n gegeben.
(2) Ist $n = 1$, so ist die Konstruktion abgeschlossen.
(3) Finde ein $p \in N_n$, sodass $x_i < x_n$ für alle $i \in N_{n-1}$, $i < p$ und $x_i > x_n$ für alle $i \in N_{n-1}$, $i \geq p$
(4) Wenn $p = 1$, so ist der linke Teilbaum von b leer.
(5) Wenn $p = n$, so ist der rechte Teilbaum von b leer.
(6) Kann man kein p finden, so handelt sich bei x nicht um

eine Postorderreihenfolge. Abbruch.

(7) Konstruiere Sequenzen $(l_i)_{i \in N_{p-1}}$, $(r_i)_{i \in N_{n-p}}$ mit

$$l_i := x_i \quad , \quad r_j := x_{j+p-1} \quad \text{für alle } i \in N_{p-1}, j \in N_{n-p}$$

(8) Wenn $p \neq 1$ dann konstruiere linken Teilbaum $b^{(l)} \in \tilde{B}_{p-1}$ aus Postorderreihenfolge $(l_i)_{i \in N_{p-1}}$ durch rekursive Anwendung dieses beschriebenen Verfahrens.

(9) Wenn $p \neq n$ dann analog zu (8) nur mit rechtem Teilbaum $b^{(r)} \in \tilde{B}_{n-p}$ und Sequenz (r_i).

Lemma: Für jede Postorderreihenfolge $p(b)$ (mit $b \in \tilde{B}_n$) ergibt der durch die Konstruktion entstehende Binärbaum $\tilde{b}$ gerade einen binären Suchbaum mit $p(\tilde{b}) = p(b)$. Des Weiteren ist $\tilde{p}$ injektiv. (Die binären Suchbäume mit paarweise verschiedenen Schlüsseln sind eindeutig durch Postorderreihenfolge bestimmt; $\tilde{b} = b$)

Beweis:

Die Konstruktion ist rekursiv, ruft sich aber selbst mit immer kleineren Sequenzlängen auf (siehe (7)-(9)). Die maximale Länge von (l_i) oder (r_i) beträgt $n-1$. Bei $n=1$ stoppt die Konstruktion. (siehe (2)) $\Rightarrow$ die maximale Rekursionstiefe beträgt n. $\Rightarrow$ Algorithmus stoppt.

Jeder Rekursionsschritt erzeugt maximal einen Knoten mit rechtem und linken Teilbaum. Durch vollständige Induktion lässt sich nun zeigen, dass diese Eigenschaft für alle Knoten gilt $\Rightarrow$ konstruiertes $\tilde{b}$ ist binärer Baum

Analog beweist man die Suchbaumeigenschaft: Wird die Konstruktion nicht abgebrochen, so wird der linke Teilbaum eines Knotens durch die Postorderreihenfolge der Elemente, die kleiner sind als der Schlüssel des Knotens, erzeugt. $\Rightarrow$ im linken Teilbaum eines jeden Knotens können nur Elemente vorkommen, die kleiner sind, als der Schlüssel des Knotens
(analog für rechten Teilbaum mit größer)
Dies ist die Eigenschaft eines Suchbaumes. $\Rightarrow$ $\tilde{b}$ ist binärer Suchbaum

Sei $\tilde{b} \in \tilde{B}_n$. Seien $p \in N_n$, $r(\tilde{b})$ der rechte Teilbaum von $\tilde{b}$ mit $n-p$ Elementen, $l(\tilde{b})$ der linke Teilbaum von $\tilde{b}$ mit $p-1$ Elementen,

$L := p(l(\bar{b}))$, $R := p(r(\bar{b}))$ und $k(\bar{b})$ der Schlüssel von $\bar{b}$.
Dann gilt nach Definition für $x := p(\bar{b})$

$$x_n = k(\bar{b}), \quad x_i := L_i \text{ für alle } i \in N_{p-1}$$
$$x_{i+p-1} = R_i \text{ für alle } i \in N_{n-p}$$

$\Rightarrow$ (verwende binäre Suchbaumeigenschaft) $x_i < x_n < x_j$
für alle $i \in N_{p-1}$, $j \in N_{n-p}$

$\Rightarrow$ ist x nun eine beliebige Postorderreihenfolge, so muss es ein $p \in N_n$ geben, sodass (3) erfüllt ist
$\Rightarrow$ für Postorderreihenfolgen wird (6) niemals ausgeführt
$\Rightarrow$ jede Postorderreihenfolge erzeugt durch die genannte Konstruktion einen binären Suchbaum

z.z.: $p(b) = p(\tilde{b})$

vollständige Induktion: $n=1$: Sequenz besteht nur aus x_1 und $\tilde{b}$ nur aus dem Schlüssel x_n.
Umgekehrt kann (x_1) nur von einem Baum mit einem Element ausgegeben werden.
$\Rightarrow p(b) = p(\tilde{b})$ (und sogar $b = \tilde{b}$)

$n \Rightarrow n+1$: Für alle $b \in \tilde{B}_k$ gilt $p(b) = p(\tilde{b})$ (und sogar $b = \tilde{b}$), $k \in N_n$
z.z.: Für alle $b \in \tilde{B}_{n+1}$ gilt $p(b) = p(\tilde{b})$ (und sogar $b = \tilde{b}$)

$$\Rightarrow p(b) = (\underbrace{p(l(b))}_{\in \tilde{B}_p}, \underbrace{p(r(b))}_{\in \tilde{B}_{n-p}}, k(b)) = x \quad \Rightarrow x_{n+1} = k(b)$$

für ein eindeutiges $p \in N_n$

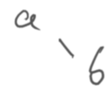
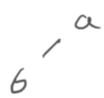
$$p(\tilde{b}) = (p(l(\tilde{b})), p(r(\tilde{b})), x_{n+1})$$

(Indukt.-vor.)
$\Rightarrow p(l(\tilde{b})) = p(l(b))$, $p(r(\tilde{b})) = p(r(b))$ (und sogar $l(\tilde{b}) = l(b)$
$r(\tilde{b}) = r(b)$)

$\Rightarrow p(\tilde{b}) = p(b) = x$ (ist linker und rechter Teilbaum von b und $\tilde{b}$ gleich, so können sie sich nur noch in $k(b)$ und $k(\tilde{b})$ unterscheiden, aber $k(b) = k(\tilde{b}) \Rightarrow b = \tilde{b}$)

$\Rightarrow \hat{p}$ ist injektiv $\square$

(b) Seien $a, b \in X$ mit $a \neq b$. Definiere

$b_1 :=$  $\qquad b_2 :=$ 

Dann sind b_1, b_2 binäre Bäume mit paarweise verschiedenen Schlüsseln und $b_1 \neq b_2$.

$$p(b_1) = (b,a) = p(b_2) \quad , \quad ip(b_1) = (b,a) = ip(b_2) \quad *$$

* ip bezeichnet die Abbildung, welche einen binären Baum auf seine inverted Postorderreihenfolge darstellt

$\Rightarrow$ die zusammengesetzte Abbildung aus p und ip ist nicht injektiv
$\Rightarrow$ ein binärer Baum ist nicht rekonstruierbar aus p und ip

Aufgabe 2

(a) Sei T ein binärer Suchbaum mit $n = 1000$ paarweise verschiedenen Schlüsseln aus $\mathbb{N}_n$. Der gesuchte Schlüssel ist $k := 424$. F sei die Sequenz der Schlüssel der passierten Knoten beim Suchen von k.

Damit F eine zulässige Sequenz ist, darf F nicht die Eigenschaften von T oder der Suche nach k verletzen.
Sei $m \in \mathbb{N}$ und $F := (f_i)_{i \in \mathbb{N}_m}$.

$\Rightarrow$ $f_1 \in \mathbb{N}_n$ ist beliebig, da er die Wurzel von T darstellt

Eigenschaft der Suche nach k: $\begin{pmatrix} f_i < k & \Leftrightarrow & f_{i+1} > f_i \\ f_i > k & \Leftrightarrow & f_{i+1} < f_i \end{pmatrix}$
für alle $i \in \mathbb{N}_{m-1}$

Der rechte Teilbaum eines Knotens, darf nur Schlüssel enthalten, die größer sind als der Schlüssel des Knotens.
Für den linken Teilbaum gilt eine analoge Bedingung nur mit „kleiner".

$$\left. \begin{array}{l} \left[f_i < f_{i+1} \Leftrightarrow \left(f_i < f_p \text{ für alle } p \in \mathbb{N}_{m-1}, p > i \right) \right] \\ \left[f_i > f_{i+1} \Leftrightarrow \left(f_i > f_p \text{ für alle } p \in \mathbb{N}_{m-1}, p > i \right) \right] \end{array} \right\} \text{für alle } i \in \mathbb{N}_{m-1}$$

Diese 4 Bedingungen führen damit direkt für alle $i \in N_{m-1}$ zu

$[f_i < k \iff (f_i < f_p \text{ für alle } p \in N_{m-1}, p > i)]$

$[f_i > k \iff (f_i > f_p \text{ für alle } p \in N_{m-1}, p > i)]$

Es reicht also diese beiden Bedingungen zu überprüfen.

a) $F := (100, 667, 513, 402, 427, 409)$

$f_1 = 100 < 424 = k$, $100 < 667, 513, 402, 427, 409$
$f_2 = 667 > 424$, $667 > 513, 402, 427, 409$
$f_3 = 513 > 424$, $513 > 402, 427, 409$
$f_4 = 402 < 424$, $402 < 427, 409$
$f_5 = 427 > 424$, $427 > 409$

$\Rightarrow$ F kann vorkommen

b) $F := (966, 513, 301, 420, 510, 407)$

$f_4 = 420 < 424$ $420 < 510, 407$ $\lightning$ denn $420 > 407$

$\Rightarrow$ F ist nicht korrekt

c) $F := (231, 666, 232, 507, 411, 419)$

analog zu a) $\Rightarrow$ F ist korrekt

d) $F := (527, 411, 417, 430, 460, 311)$

$f_5 = 430 > 424$, $430 > 460, 311$ $\lightning$ denn: $430 < 460$

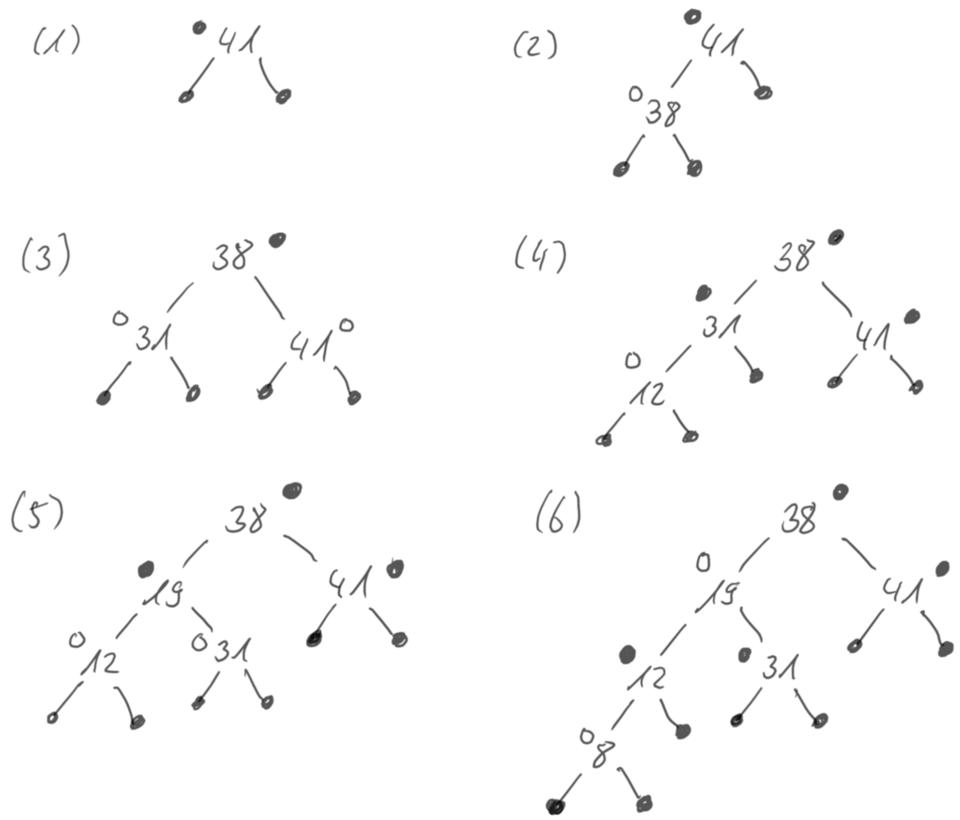
$\Rightarrow$ F nicht korrekt

(b) $F := (333, 666, 424, 511, 601, 569, 575) = (f_i)$
gesuchter Schlüssel ist $k \in \mathbb{N}$

	Intervall linker TB	Intervall rechter TB
$f_1 = 333$:	$k < 333$	$k > 333$
$f_2 = 666$:	$333 < k < 666$	$k > 666$

$f_3 = 424$:	$333 < k < 424$	$424 < k < 666$
$f_4 = 511$:	$424 < k < 511$	$511 < k < 666$
$f_5 = 601$:	$511 < k < 601$	$601 < k < 666$
$f_6 = 569$:	$511 < k < 569$	$569 < k < 601$
$f_7 = 575$:	$569 < k < 575$	$575 < k < 601$

Aufgabe 3



RB-Baum mit Schwarzhöhe (ohne NIL-Knoten) an Knoten: