AlP 3 6. Übungsblatt

36.

(7 Punkte) Wir wollen eine Variante von a-b-Bumen konstruieren, bei der weniger Speicher verschwendet wird. Es sei $b = 100$. Wie groß kann man a wählen, wenn man

- beim Entfernen bis zu *zwei* Geschwisterknoten zum Ausborgen in Betracht zieht, und
- beim Einfügen einen Geschwisterknoten zu Hilfe nimmt, falls dieser den Überlauf aufnehmen kann?

Um weniger Speicher zu verschwenden, sollen hier *möglichst grosse Werte für* a mit $b > a \geq \lfloor \frac{b+1}{2} \rfloor$ gefunden werden. Außer im Fall der Wurzel (siehe unten) hat wegen $a > 1$ jeder Knoten mindestens 1 direkten Nachbarknoten. Folgendes Verfahren ist also immer möglich:

Einfügen mit Überlauf

Falls der linke oder der rechte direkte Nachbarknoten weniger als b Einträge hat, gibt der Überlaufknoten den kleinsten Eintrag an den linken oder den grössten Eintrag an den rechten direkten Nachbarn ab.

Falls jeder direkte Nachbarknoten voll ist, also *schon* b *Einträge hat* , werden die b Einträge eines vollen direkten Nachbarknotens mit den $b+1$ Einträgen des übergelaufenen Knotens (sortiert) auf 3 minimal gefüllte neue Knoten mit je mindestens a Einträgen *aufgeteilt.* (Die beiden alten Knoten werden durch die 3 neuen ersetzt.)

Damit das immer möglich ist, muß die Anzahl Einträge in jedem der 3 neuen Knoten zwischen a und b liegen. Insbesondere der am wenigsten gefüllte Knoten (von 3 möglichst gleichmäßig - also mit $\lfloor \frac{2b+1}{3} \rfloor$ und $\lceil \frac{2b+1}{3} \rceil$ Einträgen - gefüllten Knoten) muß mindestens a Einträge haben. Es muß also gelten

$a \leq \lfloor \frac{2b+1}{3} \rfloor \leq \lceil \frac{2b+1}{3} \rceil \leq b \quad \Rightarrow \quad a \leq \lfloor \frac{2b+1}{3} \rfloor$

Das Elter des übergelaufenen Knotens hat so $1 = 3 - 2$ Einträge mehr als vorher, evt. also auch einen Überlauf, der nach denselben Regeln behandelt wird.

Für $b = 100$ gilt dann, daß

$$a \leq \left\lfloor \frac{2 * 100 + 1}{3} \right\rfloor = \frac{201}{3} = 67$$

d.h. a darf höchstens 67 sein.

Entfernen eines Eintrags mit Unterlauf

Falls der linke oder der rechte direkte Nachbarknoten mindestens $a + 1$ Einträge hat, erhält der Knoten mit Unterlauf von seinem linken Nachbarn den maximalen oder vom rechten Nachbarn dessen kleinsten Eintrag.

Falls beide direkten Nachbarn (genau) a Einträge haben, aber ein direkter Nachbar dieser Nachbarn mindestens $a + 1$ Einträge hat, gibt dieser Nachbar (2.Art) seinen kleinsten bzw. größten Eintrag an den gemeinsamen Nachbarn des Unterlauf-Knotens und dieser gemeinsame Nachbar mit dann $a + 1$ Einträgen gibt wie im vorigen Fall einen Eintrag an den Unterlauf-Knoten ab.

Falls die zwei nächsten Nachbarn - auf beiden Seiten - des Unterlaufknotens nur je a Einträge haben, so werden die $a - 1$ Einträge des Unterlauf-Knotens mit den $2a$ Einträgen seiner 2 direkten Nachbarn oder 2 nächsten Nachbarn zur Linken oder zur Rechten sortiert in 2 neuen Knoten zusammengefasst / verschmolzen. (Diese 2 „neuen“ Knoten ersetzen die ursprünglichen 3 Knoten.) Die beiden neuen Knoten dürfen dabei nur zwischen a und b Einträge erhalten. Damit das immer möglich ist muß gelten

$2a \leq 2a + (a - 1) = 3a - 1 \leq 2b \quad \Rightarrow \quad 1 \leq a \leq \lceil \frac{2b+1}{3} \rceil$

(Der im letzten Fall beim Elter des Unterlauf-Knotens evt. auftretende Unterlauf wird genauso behandelt.)

Für $b = 100$ muß

$$a \leq \left\lceil \frac{2 * 100 + 1}{3} \right\rceil = \frac{201}{3} = 67$$

d.h. a darf wie beim Einfügen höchstens 67 sein.

Sonderfall Wurzel

Die Wurzel hat *keine(n)* Nachbarknoten die *beim Einfügen mit Überlauf* einen Eintrag übernehmen könnten. Bei den üblichen a-b-Bäumen gilt schon die Sonderregel, daß die Wurzel zwischen 1 und b Einträge haben darf. Damit sich ein Überlauf der Wurzel immer aufteilen lässt, und zwar auf eine neue Wurzel mit 2 Kindern, die jeweils zwischen a und b Einträge haben, erlaubt man folgende **zusätzliche Sonderregel**:

> Die Wurzel darf bis zu $\max(b, 2a - 1)$ hier also bis zu $2a - 1$ Einträge enthalten, da $a \geq \lceil \frac{b+1}{2} \rceil$ sein soll.

Mit dem oben für $b = 100$ gewonnen größtmöglichen Wert $a = 67$ ergibt sich, daß die Wurzel zwischen 1 und $133 = 2 * 67 - 1 = 2a - 1$ Einträgen haben darf.

37.

(9 Punkte) *Rot-Schwarz-Bäume* sind binäre Bäume mit folgenden Eigenschaften:

(a) Jeder innere Knoten hat zwei Kinder.

(b) Jeder Knoten ist entweder als *rot* oder als *schwarz* gekennzeichnet.

(c) Die Wurzel und alle Blätter sind schwarz.

(d) Die Kinder eines roten Knotens sind schwarz.

(e) Ein schwarzer Knoten kann höchstens ein rotes Kind haben.

(f) Alle Wege von der Wurzel zu den Blättern enthalten gleich viele schwarzen Knoten.

Zeichnen Sie Rot-Schwarz-Bäume mit 5, 7 und 12 Blättern. Zeigen Sie, dass man aus jedem Rot-Schwarz-Baum einen 2-3-Baum machen kann, und umgekehrt.

Eigenschaften eines 2-3 Baums:

1. Alle Wurzel-Blatt Pfadlängen sind gleich (=Höhe des Baums)
2. jeder innere Knoten hat entweder 1 Schlüssel(s1) und 2 Kinder(t0,t1) oder 2 Schlüssel(s1,s2) und 3 Kinder(t0,t1,t2)

($\Leftarrow$) Aus jedem 2-3-Baum kann man einen Rot-Schwarz-Baum machen:

Alle Knoten werden zunächst schwarz eingefärbt. Wegen 1. sind alle Wurzel-Blatt Pfadlängen gleich. Knoten mit 2 Schluesseln (s1,s2) und 3 Kindern (t0,t1,t2), werden derart verändert, dass s1 mit t0 und t1 als Kinder zum neuen linken roten Kind des schwarz bleibenden, nun nur noch s2 als Schlüssel habenden Knotens gemacht. t2 bleibt dessen rechtes schwarzes Kind. s2 hat also ein linkes rotes Kind (s1) und ein rechtes schwarzes (t2).(Analog, wäre auch möglich: aus s1,s2 wird s1 mit linkem schwarzem kind t0 und rechtem neuen roten Knoten s2. s2 hat dann die schwarzen Kinder t1 und t2)

Der neu entstandene Knoten s1, der die Pfadlänge vergrössert, wird rot eingefärbt. Dies wird bei allen Knoten mit 2 Schlüsseln gemacht. Knoten mit nur einem Schlüssel bleiben unverändert schwarz.

- Damit wird sicher gestellt, dass lediglich alle neu entstandenen und damit den Wurzel-Blatt-Weg verlängernden Knoten rot eingefärbt sind.($\Rightarrow$ (f), (b) erfüllt)
- Wegen 2. und dem Verändern von ausschliesslich Knoten mit 2 Schlüsseln, wie vorhin beschrieben ($\Rightarrow$ (e), (d)).
- Da es auf Blattebene keine Knoten mit 2 Schlüsseln gibt und jeder Knoten mit s1,s2 nach dem Umbau (s2) schwarz bleibt ($\Rightarrow$ (c))
- Aus jedem Knoten mit 2 Schlüsseln (s1,s2) und 3 Kindern(t0,t1,t2) werden zwei Knoten mit je einem Schlüssel und jeweils 2 Kindern. Aus s1,s2 wird s2, mit den Kindern s1 und t2. s1 hat t0 und t1 als Kinder ($\Rightarrow$ (a))

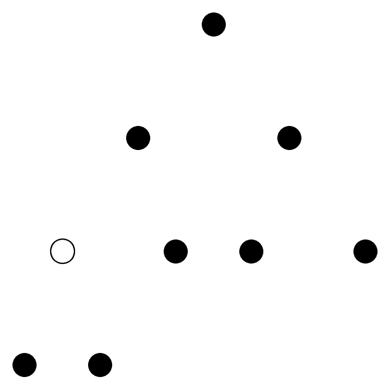
($\Rightarrow$) Aus jedem Rot-Schwarz-Baum kann man einen 2-3-Baum machen:

- Alle Knoten mit einem schwarzen und einem roten Kind werden verändert. Falls der rote Knoten das linke Kind (s1) ist, wird sein Elternknoten s zum Knoten s1,s erweitert, mit den Kindern t0,t1 (Kinder des roten Knotens, sind schwarz (d)) und t2 (war vorher schon schwarzes rechtes Kind von s). Ist der rote Knoten(s2) das rechte Kind von s, so wird s zu s,s2 mit den Kindern t0 (vorher schon linkes schwarzes Kind von s) sowie t1,t2 (Kinder des roten Knotens s2, ebenso schwarz(d)).

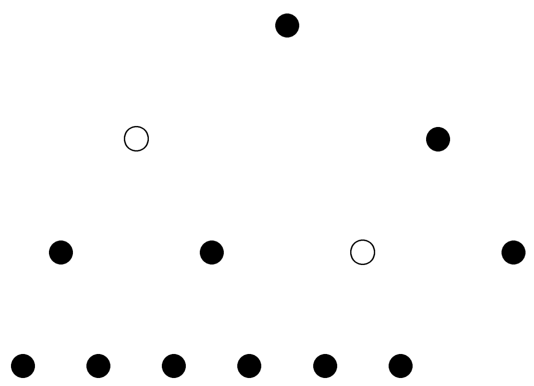
  Da auf diese Weise alle schwarzen Knoten mit einem roten Kind um dessen Schlüssel erweitert werden (shwarze Knoten mit zwei schwarzen Kindern werden nicht verändert) und somit alle roten Knoten verschwinden, bleiben nur schwarze Knoten übrig. Wegen (f), (b) ist Eigenschaft 1. erfüllt.
- Mit (a),(c),(d) und (e) wird sichergestellt, dass der entstehende 2-3-Baum weder einen Über- noch einen Unterlauf hat. Jeder Knoten mit einem Schlüssel(s) und je zwei Kindern kann damit um höchstens einen weiteren Schlüssel (s2 oder s1, je nachdem, ob der rote Knoten rechtes oder linkes Kind ist) wie oben beschrieben, erweitert werden. Alle erweiterten Knoten haben damit 2 Schlüssel und 3 Kinder. Wegen (c) gilt zum einen, dass jeder rote Knoten genau 2 schwarze Kinder hat (Blätter, bei denen das nicht zuträfe, da keine Kinder, sind alle schwarz) damit entsteht kein Unterlauf und zum anderen, ist die Wurzel auch schwarz, der Umbau, wie oben beschrieben, kann also konsequent bei allen Knoten durchgeführt werden.

  Damit gilt auch 2.

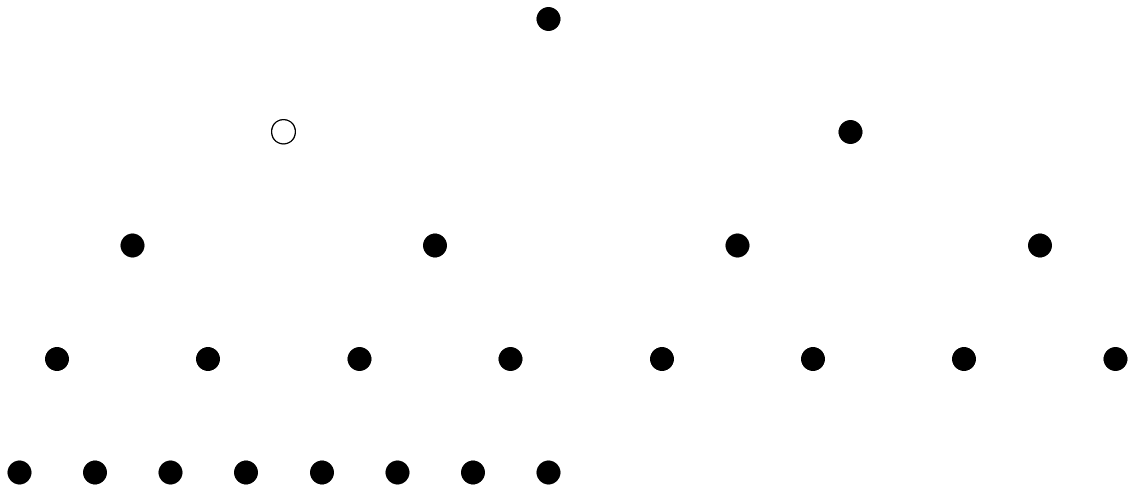
Rot-Schwarz-Baum mit 5 Blättern:



Rot-Schwarz-Baum mit 7 Blättern:



Rot-Schwarz-Baum mit 12 Blättern:



38.

(4 Punkte) Es sei A_h die kleinstmögliche Anzahl von Blättern eines AVL-Baumes mit Höhe h. Beweisen Sie, dass die Zahlen A_h die Fibonacci-Zahlen sind.

Was können Sie daraus für die kleinstmögliche Anzahl $n(h)$ von *inneren Knoten* eines AVL-Baumes mit Höhe h schließen?

Vor.: Sei binärer Baum t_h mit $h \in N$ rekursiv wie folgt definiert:

t_0 sei der Baum, der aus genau 1 Knoten, d.h. nur 1 Blatt besteht,
t_1 sei der Baum, der aus Wurzelknoten und 1 Blatt als linkem Kind besteht und
$\forall_{2 \geq h \in N}$ t_h sei der Baum, der aus Wurzelknoten mit t_{h-1} als linkem Kind und t_{h-2} als rechtem Kind besteht.
Bezeichne $B(t_h)$ die Anzahl Blätter und $K(t_h)$ die Anzahl Knoten von t_h.

Beh.: $\forall_{h \in N}$ t_h ist - bis auf Isomorphie - ein AVL-Baum der Höhe h mit kleinster Anzahl Blätter (Knoten) und $B(t_h) = \text{Fib}(h+1) \wedge K(t_h) = \text{Fib}(h+3) - 1$.
Isomorph heißt hier, daß Bäume als gleich bzw. isomorph angesehen werden, die durch beliebige Vertauschung von linkem und rechten Kind/Unterbaum in inneren Knoten ineinander übergehen.

Bew.: (Induktion über Höhe h)
t_0 bzw. t_1 sind die kleinstmöglichen Bäume der Höhe 0 bzw. 1 und erfüllen die Bedingung für AVL-Bäume (, daß die Höhen der Kinder /Unterbäume von inneren Knoten sich um höchstens 1 unterscheiden).
t_0 hat
1 Blatt also ist $B(t_0) = 1 = \text{Fib}(1) = \text{Fib}(0+1)$ u.
1 Knoten d.h. $K(t_0) = 1 = 2 - 1 = \text{Fib}(0+3) - 1$.
t_1 hat
1 Blatt also ist $B(t_0) = 1 = \text{Fib}(2) = \text{Fib}(1+1)$ u.
2 Knoten d.h. $K(t_1) = 2 = 3 - 1 = \text{Fib}(1+3) - 1$.
Somit sind t_0 bzw. t_1 AVL-Bäume der Höhe 0 bzw. 1 mit der kleinstmöglichen Anzahl Blätter(Knoten) und erfüllen die Aussagen über $B(t_h)$ und $K(t_h)$.
(Ließe man irgendeinen Knoten in t_0 bzw. t_1 weg, hätte man keinen Baum der Höhe 0 bzw. 1, erst recht also keinen AVL-Baum der Höhe 0 bzw. 1. t_0 bzw. t_1 sind also minimal.)

Sei also schon bis zur Höhe $h \geq 1$ gezeigt, daß t_h AVL-Baum der Höhe h mit minimaler Anzahl Blätter(Knoten) ist, $B(t_h) = \text{Fib}(h+1)$ und $K(t_h) = \text{Fib}(h+3) - 1$.
Ein AVL-Baum der Höhe $h+1$ muß

- als Baum der Höhe $h+1$ (wegen $h \geq 1$) eine Wurzel mit größtem Unterbaum der Höhe h haben,
- wegen der zusätzlichen Bedingung für AVL-Bäume noch einen 2. Unterbaum der Höhe h oder $h-1$.
- Beide Unterbäume müssen dabei selbst AVL-Bäume sein.

Da ein AVL-Baum der Höhe $h-1$ mit kleinstmöglicher Anzahl Blätter offensichtlich *höchstens soviele* Blätter(aber nicht soviele Knoten) hat, wie ein ebensolcher AVL-Baum der Höhe h, hat der 2. Unterbaum die Höhe $h-1$. Ein AVL-Baum der Höhe $h+1$ mit minimaler Anzahl Blätter(Knoten) muß bei $h \geq 1$ also bestehen aus

- Wurzelknoten mit je 1
- AVL-Unterbaum der Höhe h mit minimaler Anzahl Blätter(Knoten),
- AVL-Unterbaum der Höhe $h-1$ mit minimaler Anzahl Blätter(Knoten).

(Weglassen von Knoten in einem Unterbaum führt bestenfalls zu Unterbäumen, die keine AVL-Bäume sind oder zu geringe Höhe haben, wegen der vorausges. Minimalität bei den AVL-Unterbäumen.)
Somit ist ein AVL-Baum der Höhe $h+1$ mit minimaler Anzahl Blätter(Knoten) - bis auf Isomorphie - gleich t_{h+1}, da nach IV. AVL-(Unter)bäume der Höhe h bzw. $h-1$ mit minimaler Anzahl Blätter(Knoten) gerade t_h bzw. t_{h-1} sind.

Nach Konstruktion von t_{h+1} besteht dieser aus 1 Wurzel (also *nicht* Blatt) mit t_h und t_{h-1} als Unterbäumen. Für die Anzahl Blätter(Knoten) gilt also

$$\begin{aligned} B(t_{h+1}) &= 0 + B(t_h) + B(t_{h-1}) \\ &\overset{\text{IV.}}{=} \text{Fib}(h+1) + \text{Fib}(h) \\ &= \text{Fib}(h+2) \\ &= \text{Fib}((h+1)+1) \end{aligned}$$

$$\begin{aligned} K(t_{h+1}) &= 1 + K(t_h) + K(t_{h-1} \\ &\overset{\text{IV.}}{=} 1 + (\text{Fib}(h+3) - 1) + (\text{Fib}((h+2) - 1) \\ &= (\text{Fib}(h+3) + \text{Fib}(h+2)) - 1 - 1 + 1 \\ &= \text{Fib}(h+4) - 1 \\ &= \text{Fib}((h+1)+3) - 1 \end{aligned}$$

Damit gilt die Beh. mit $h \geq 1$ auch für $h+1$, insgesamt also auf N.

Daraus ergeben sich sofort die gesuchten **Resultate:**

Da die t_h gerade AVL-Bäume der Höhe h mit kleinstmöglicher Anzahl Blätter A_h für $h \in N$ sind, gilt damit $\forall_{h \in N}$ $A_h = B(t_h) = \text{Fib}(h+1)$, die A_h sind also die Fibonacci-Zahlen (ab $1 = \text{Fib}(1)$).

Die kleinstmögliche Anzahl innerer Knoten bei einem AVL-Baum der Höhe h - also $n(h)$ - ist gerade die Differenz aus der minimalen Anzahl Knoten und der minimalen Anzahl Blätter bei einem AVL-Baum der Höhe h, d.h.

$$\begin{aligned} n(h) &= K(t_n) - B(t_h) \\ &= (\text{Fib}(h+3) - 1) - \text{Fib}(h+1) \\ &= ((\text{Fib}(h+2) + (\text{Fib}(h+1)) - 1) - \text{Fib}(h+1) \\ &= (\text{Fib}(h+2) - 1 \end{aligned}$$

Für die minimalen AVL-Bäume der Höhe $h \in \{0, 1, \ldots, 7\}$ ergibt sich also

h	0	1	2	3	4	5	6	7
Fib(h)	0	1	1	2	3	5	8	13
$B(h)$	1	1	2	3	5	8	13	21
$n(h)$	0	1	2	4	7	12	20	33
$K(h)$	1	2	4	7	12	20	33	54