Lehrstuhl für Informatik I
— Informationsstrukturen und wissensbasierte Systeme —
Universität Würzburg

Prof. Dr. H. Noltemeier

Würzburg, 12. Juni 1998

Übungen zur Vorlesung Praktische Informatik II (SS98)

Musterlösung zum 2. Übungsblatt

Lösung zu Aufgabe 6: (B'-Baum)

Sei $T = (V, R)$ ein Wurzelbaum. T heißt B'-Baum der Ordnung k genau dann, wenn gilt:

i) Alle Blätter besitzen die gleiche Höhe.

ii) Die Wurzel von T hat entweder keinen Sohn oder mindestens 2 Söhne und höchstens $2k$ Söhne.

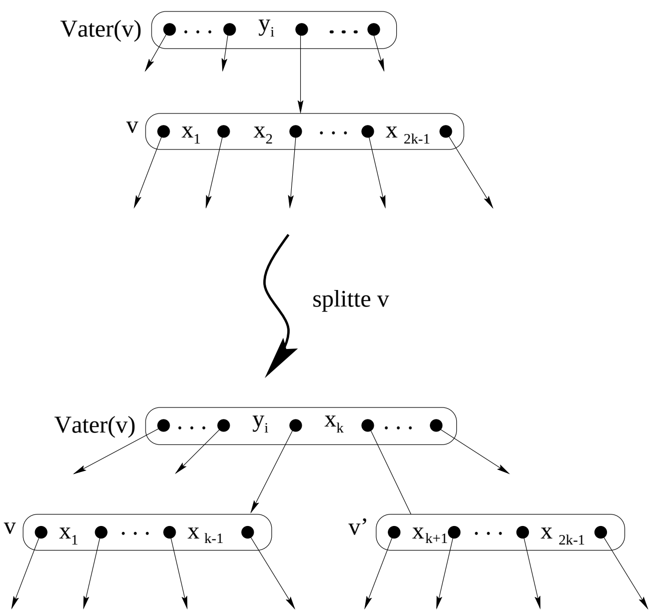
iii) Ist v eine Ecke und weder Wurzel noch Blatt, so besitzt v mindestens k und höchstens $2k$ Söhne.

Es ist ein Einfügealgorithmus für einen B'-Baum anzugeben, der den Pfad von der Wurzel zum Knoten nur einmal (von oben nach unten) durchläuft; d.h. der Einfügealgorithmus restrukturiert den neuen Baum bereits während der top-down-Suchphase.

Nach der Definition des B'-Baumes befinden sich in jeder Page (mit Ausnahme der Wurzel) mindestens $k-1$ und höchstens $2k-1$ Schlüsselwerte. Diese Eigenschaft erlaubt folgenden Einfügealgorithmus:

Algorithmus 1 Einfügen des Schlüssels s in einen B'-Baum

1: Durchlaufe den Baum entlang des Suchpfades zu s, beginnend bei der Wurzel:
2: Befinden sich in der Wurzel $2k-1$ Elemente, so teile diese Elemente in die $k-1$ kleineren, das mittlere und die $k-1$ größeren Elemente auf. Das mittlere Element bildet die neue Wurzel, welche auf die beiden Pages mit den $k-1$ kleineren und den $k-1$ größeren Elementen zeigt.
3: Durchlaufe nun alle übrigen Pages entlang des Suchpfades einschließlich des Blattes und splitte sie, falls sie $2k-1$ Schlüsselwerte enthalten. Das mittlere Element wird dann jeweils in den Vaterknoten verschoben. (Dies ist grundsätzlich möglich, da bereits sichergestellt wurde, daß alle Knoten von der Wurzel bis zum aktuellen nicht voll besetzt sind.
4: Anschließend kann man den Schlüssel s im evtl. gesplitteten Blatt einfügen.



Lösung zu Aufgabe 7: (Paralleler Zugriff auf B-Bäume)

Zum parallelen Zugriff in einem B-Baum ist es notwendig, Teile des Baums zu sperren, falls die Möglichkeit besteht, daß sich noch nicht beendete Einfüge- oder Löschoperationen auf diese Teile durch Knotenteilungen oder Knotenverschmelzungen auswirken. Suchoperationen sind weitgehend unkritisch, hier muß nur darauf geachtet werden, daß Teile des Baums, die sich noch in der Umstrukturierung befinden, nicht betreten werden.

Ein naiver Ansatz bestünde daher darin, eine beliebige Anzahl von Suchoperationen parallel ablaufen zu lassen, während beim Einfügen oder Löschen stets der ganze Baum gesperrt wird.

Da dies aber vor allem bei häufiger Änderung des Baumes kaum eine hohe Zahl gleichzeitiger Zugriffe garantiert, kann eine der folgenden Lösungen Verwendung finden:

Lösung 1: Es sind zwei Arten von Flags notwendig, ein „ρ-lock“ und ein „ξ-lock“. Ist in einem Knoten bereits ein ρ-lock gesetzt, kann in diesem Knoten nicht auch noch ein ξ-lock gesetzt werden und umgekehrt. Allerdings können für einen Knoten mehrere ρ-locks gesetzt werden. Es wird in dieser Lösung und auch in den folgenden stets davon ausgegangen, daß die locks in der Reihenfolge ihrer Anforderung gesetzt werden, d.h. wenn lock a früher als lock b angefordert wurde, wird zuerst lock a gesetzt (eventuell muß gewartet werden, bis ein anderer lock aufgehoben wird, damit lock a gesetzt werden kann!). Hiermit ergibt sich dann die in Algorithmus 2 beschriebene Vorgehensweise für den Suchzugriff.

Ein Knoten soll als *sicher* gelten, falls sich Restrukturierungsmaßnahmen im Teilbaum mit diesem Knoten als Wurzel nicht auf seine Vorfahren auswirken können. Beim Einfügen ist ein Knoten also *sicher*, falls er weniger als $2k$ Schlüssel enthält. Für das Löschen muß er wenigstens $k+1$ Schlüssel enthalten. Hiermit erhält man dann die Beschreibung für einen Einfüge- oder Löschzugriff (siehe Algorithmus 3).

Algorithmus 2 Suchzugriff

```
Sperre die Wurzel durch einen ρ-lock
Mache die Wurzel zum aktuellen Knoten
while Der aktuelle Knoten enthält nicht den gesuchten Schlüssel und ist kein Blatt do
    Sperre den zu durchsuchenden Sohn des aktuellen Knoten durch einen ρ-lock
    Hebe den ρ-lock für den aktuellen Knoten wieder auf
    Mache den zu durchsuchenden Sohn des aktuellen Knoten zum aktuellen Knoten
```

Algorithmus 3 Einfüge- oder Löschzugriff

```
Sperre die Wurzel durch einen ξ-lock.
Mache die Wurzel zum aktuellen Knoten
while Der aktuelle Knoten ist kein Blatt do
    Sperre den passenden Sohn des aktuellen Knoten durch einen ξ-lock
    Mache den Sohn zum aktuellen Knoten
    if Der aktuelle Knoten ist sicher then
        Hebe alle ξ-locks längs des Pfades zum aktuellen Knoten wieder auf
```

Beim Löschen ist zu beachten, daß dies so nur funktionieren kann, wenn sich der zu löschende Schlüssel in einem Blatt befindet. Ist dies nicht der Fall, so ist eine kleine Modifikation notwendig, wie sie in Algorithmus 4 beschrieben ist.

Da beim Löschen des am weitesten rechts stehenden Schlüssels im linken Teilbaum das normale Löschverfahren verwendet wird, ist es ausgeschlossen, daß hierbei eine Störung eines bereits vorher angefangenen Suchzugriffs erfolgt. Um die Sperrung des Knotens, der den eigentlich zu löschenden Schlüssel enthält, früher aufzuheben, muß man nur im Schritt 10 – sobald der passende Ersatzschlüssel gefunden ist – zuerst die Schlüssel vertauschen, den Knoten, falls der entsprechende Sohn *sicher* ist, wieder freigeben und erst dann die notwendigen Restrukturierungsmaßnahmen ausgehend vom Blatt durchführen.

Bei der eben vorgestellten Methode zum parallelen Zugriff auf B-Bäume kann es allerdings vorkommen, daß trotzdem beim Einfügen und Löschen sehr große Teile des Baums gesperrt werden, obwohl nur ein kleiner Teil tatsächlich verändert wird. Deshalb kann man beim Einfügen und Löschen (in einem Blatt) zunächst suchend auf den Baum zugreifen. Wurde dann der zu modifiziernde Knoten gefunden, sperrt man diesen durch einen ξ-lock. Ist der Knoten unsicher, so wiederholt man den Zugriff diesmal als echte Einfüge- oder Löschoperation. Beim Löschen in einem inneren Knoten ergeben sich ebenfalls keine großen Unterschiede.

Lösung 2 Bei diesem Ansatz verwendet man drei Flags. Diese sind ein „ρ-lock“, ein „ξ-lock“ und ein „α-lock“. Für ρ- und ξ-lock gilt die Beziehung wie in Lösung 1, ein ρ-lock kann trotz gesetztem α-lock gesetzt werden und umgekehrt, und ein α-lock kann nicht gleichzeitig mit einem ξ-lock auftreten, ein α-lock kann aber in einen ξ-lock umgewandelt werden.

Die Vorgehensweise für den *Suchzugriff* bleibt unverändert. Das *Einfügen* und *Löschen* ist wie in Algorithmus 5 möglich.

Algorithmus 4 Löschzugriff (Schlüssel nicht im Blatt)

1: Sperre die Wurzel durch einen ξ-lock.
2: Mache die Wurzel zum aktuellen Knoten
3: **while** Der aktuelle Knoten enthält nicht den zu löschenden Schlüssel und ist kein Blatt **do**
4:   Sperre den passenden Sohn des aktuellen Knoten durch einen ξ-lock
5:   Mache den Sohn zum aktuellen Knoten
6:   **if** Der aktuelle Knoten ist sicher **then**
7:     Hebe alle ξ-locks längs des Pfades zum aktuellen Knoten wieder auf
8: **if** Der zu löschende Schlüssel wurde gefunden **then**
9:   Hebe alle ξ-locks längs des Pfades zum aktuellen Knoten wieder auf und sperre den aktuellen Knoten durch einen ξ-lock und hebe die Sperrung dieses Knotens in Schritt 10 nicht auf
10:   Lösche unter Anwendung des Verfahren zum Löschen in einem Blatt den am weitesten rechts stehenden Schlüssel im linken Teilbaum und setzte ihn an die Stelle des eigentlich zu löschenden Schlüssels
11:   Hebe den ξ-lock für den aktuellen Knoten wieder auf

Algorithmus 5 Einfügen und Löschen

1: Sperre die Wurzel durch einen α-lock
2: Mache die Wurzel zum aktuellen Knoten
3: **while** Der aktuelle Knoten ist kein Blatt **do**
4:   Sperre den passenden Sohn des aktuellen Knoten durch einen α-lock
5:   Mache den Sohn zum aktuellen Knoten
6:   **if** Der aktuelle Knoten ist sicher **then**
7:     Hebe alle α-locks längs des Pfades zum aktuellen Knoten wieder auf.
{Nun wurde ein Blatt erreicht}
8: **if** Der zu löschende oder einzufügende Schlüssel befindet sich im Blatt **then**
9:   Wandle von oben nach unten alle α-locks in ξ-locks um und beginne mit der Umstrukturierung

Befindet sich der zu löschende Schlüssel in einem inneren Knoten oder der Wurzel, so modifiziert man die Vorgehensweise analog Lösung 1. Hierbei muß allerdings in Schritt 10 unbedingt Algorithmus 3 verwendet werden, da sonst die Gefahr besteht, daß ein ρ-lock von einem α-lock „überholt" wird.

Diese Lösung hat also den Vorteil, daß während eines laufenden Einfüge- oder Löschzugriffs zunächst noch im Baum gesucht werden kann, und daß dann nur die wirklich zu verändernden Knoten endgültig gesperrt werden.

Die Lösungen orientieren sich weitgehend an dem Artikel „Concurrency of Operations on B-Trees", Acta Informatica 9, 1–21 (1997). Dort ist auch eine kombinierte Lösung zu finden.

Lösung zu Aufgabe 8: (B*-Bäume)

Es soll angegeben werden, wieviele Schlüsselwerte maximal in einem B*-Baum vom Typ $(k, k_l, \cdot)$ der Höhe h ($h \in \mathbb{N}_+$) repräsentiert werden können.

Für einen B*-Baum vom Typ (k, k_l, h) gilt

$$|S^{B^*}_{max}| = (2k+1)^h \cdot 2k_l$$

da in jeder Höhenstufe $2k+1$ Söhne pro Knoten möglich sind und jedes Blatt maximal $2k_l$ Schlüsselwerte enthalten kann.

Wir vergleichen nun die Höhen eines B-Baumes vom Typ $(k, \cdot)$ und eines B*-Baumes vom Typ $(k, k, \cdot)$. Aus obiger Abschätzung und der Tatsache, daß ein maximal besetzter B-Baum der Höhe h genau

$$\begin{aligned} |S^B_{max}| &= 2k \cdot \sum_{i=0}^{h} (2k+1)^i \\ &= 2k \cdot \frac{(2k+1)^{h+1} - 1}{(2k+1) - 1} \\ &= (2k+1)^{h+1} - 1 \end{aligned}$$

Schlüssenwerte enthält, leitet sich ab, daß gilt:

$$|S^{B^*}_{max}(h)| < |S^B_{max}(h)| < |S^{B^*}_{max}(h+1)| < |S^B_{max}(h+1)|$$

Hieraus erhält man, daß für eine beliebige Schlüsselwertmenge S mit n Elementen gilt:

$$\begin{aligned} n \in [|S^{B^*}_{max}(h)| + 1, |S^B_{max}(h)|] \quad &\Rightarrow \quad \exists\ B\text{-Baum der Höhe } h, \text{ der } S \text{ repräsentiert,} \\ &\qquad \text{sowie ein } B^*\text{-Baum der Höhe } h+1 \\ n \in [|S^B_{max}(h)| + 1, |S^{B^*}_{max}(h+1)|] \quad &\Rightarrow \quad \exists\ B\text{-Baum der Höhe } h+1, \text{ der } S \text{ repräsentiert,} \\ &\qquad \text{sowie ein } B^*\text{-Baum der Höhe } h+1 \\ \Rightarrow \quad h_{B^*} - h_B \in \{0, 1\} \end{aligned}$$

Also gibt es je eine Repräsentation von S, so daß sich die Höhe der Bäume höchstens um eins unterscheiden.

Lösung zu Aufgabe 9: (Eigenschaften von Segmentbäumen)

(a) **Zu zeigen:**

Für die Höhe eines Segmentbaumes gilt: $h_{ST(l,r)} = \lceil \log_2(r-l) \rceil$

Beweis:

Vollständige Induktion über $(r-l)$.

(IA) Sei $(r-l) = 1$. Dann besteht $ST(l,r)$ aus einem einzigen Knoten, der zugleich Wurzel und Blatt ist, und es gilt: $h_{ST(l,r)} = 0 = \lceil \log_2 1 \rceil = \lceil \log_2(r-l) \rceil$.

(IV) Es sei $h_{ST(l,r)} = \lceil \log_2(r-l) \rceil$ für alle Segmentbäume $ST(l,r)$ mit $(r-l) \leq c$ für ein $c \geq 1$ bereits gezeigt.

(IS) $c \to c+1$. Sei $(r-l) = c+1$.
Dann folgt mit (IV):

$$\begin{aligned} h_{ST(l,r)} &= 1 + \max\left\{ \left\lceil \log_2\left(\left\lfloor \frac{r+l}{2} \right\rfloor - l \right) \right\rceil, \left\lceil \log_2\left(r - \left\lfloor \frac{r+l}{2} \right\rfloor \right) \right\rceil \right\} \\ &= \max\left\{ \left\lceil 1 + \log_2 \left\lfloor \frac{r-l}{2} \right\rfloor \right\rceil, \left\lceil 1 + \log_2 \left\lceil \frac{r-l}{2} \right\rceil \right\rceil \right\} \\ &= \left\lceil 1 + \log_2 \left\lceil \frac{r-l}{2} \right\rceil \right\rceil \end{aligned}$$

Fall 1: $(r-l)$ ist gerade.
Dann folgt weiter:

$$\begin{aligned} h_{ST(l,r)} &= \left\lceil 1 + \log_2\left(\frac{r-l}{2} \right) \right\rceil \\ &= \lceil \log_2(r-l) \rceil \end{aligned}$$

Fall 2: $(r-l)$ ist ungerade.
Dann folgt weiter:

$$\begin{aligned} h_{ST(l,r)} &= \left\lceil 1 + \log_2\left(\frac{r-l+1}{2} \right) \right\rceil \\ &= \lceil \log_2(r-l+1) \rceil \end{aligned}$$

Wegen der Monotonie des Logarithmus gilt

$$\lceil \log_2(r-l) \rceil \leq \lceil \log_2(r-l+1) \rceil$$

Außerdem gilt $\lceil \log_2(r-l) \rceil < \lceil \log_2(r-l+1) \rceil$ gdw. $r-l = 2^n$ für ein $n \in \mathbb{N}$. Da aber $(r-l)$ als ungerade und größer als 1 vorausgesetzt wurde, kann $(r-l)$ keine Zweierpotenz sein, und es gilt Gleichheit:

$$h_{ST(l,r)} = \lceil \log_2(r-l+1) \rceil = \lceil \log_2(r-l) \rceil$$

Also gilt in allen Fällen die Behauptung.

□

(b) **Zu zeigen:**

Sind v, v' zwei Blätter in $ST(l,r)$, so gilt: $|h(v) - h(v')| \leq 1$, wobei $h(x)$ die Höhe des Knotens x in $ST(l,r)$ bezeichnet.

Beweis:

Nach Teilaufgabe (a) hat jedes Blatt v aus $ST(l,r)$ eine Höhe $h(v) \leq \lceil \log_2(r-l) \rceil$.

Für jeden Segmentbaum $ST(l,r)$ mit linkem Teilbaum $T_{links} = ST(l, \lfloor \frac{r+l}{2} \rfloor)$ und rechtem Teilbaum $T_{rechts} = ST(\lfloor \frac{r+l}{2} \rfloor, r)$ gilt folgende Beziehung für die Höhen der Teilbäume:

$$\begin{aligned}
h_{T_{links}} &= \left\lceil \log_2 \left(\left\lfloor \frac{r+l}{2} \right\rfloor - l \right) \right\rceil \\
&\leq \left\lceil \log_2 \left(\frac{r+l}{2} - l \right) \right\rceil \\
&= \left\lceil \log_2 \left(\frac{r-l}{2} \right) \right\rceil \\
&= \left\lceil \log_2 \left(r - \frac{r+l}{2} \right) \right\rceil \\
&\leq \left\lceil \log_2 \left(r - \left\lfloor \frac{r+l}{2} \right\rfloor \right) \right\rceil \\
&= h_{T_{rechts}}
\end{aligned}$$

Das heißt der rechte Teilbaum ist mindestens so hoch wie der linke. Wendet man diese Beobachtung rekursiv auf alle Teilbäume an, so folgt:

(i) Das Blatt größter Höhe ist das rechteste Blatt im Baum.

(ii) Das Blatt geringster Höhe ist das linkeste Blatt im Baum.

Per vollständiger Induktion über $(r-l)$ wird gezeigt, daß das linkeste Blatt im Baum die Höhe $h_l = \lfloor \log_2 (r-l) \rfloor$ hat:

(IA) Sei $(r-l) = 1$. Dann besteht $ST(l,r)$ aus einem einzigen Knoten, der zugleich Wurzel und Blatt ist, und es gilt: $h_l = 0 = \lfloor \log_2 1 \rfloor = \lfloor \log_2 (r-l) \rfloor$

(IV) Es sei $h_l = \lfloor \log_2 (r-l) \rfloor$ für alle Segmentbäume $ST(l,r)$ mit $(r-l) \leq c$ für ein $c \geq 1$ bereits gezeigt.

(IS) $c \to c+1$. Sei $(r-l) = c+1$.

Fall 1: $(r-l)$ ist gerade.

Dann ist auch $r+l = (r-l) + 2l$ gerade, und mit (IV) folgt weiter:

$$\begin{aligned}
h_l &= 1 + \left\lfloor \log_2 \left(\frac{r+l}{2} - l \right) \right\rfloor \\
&= \log_2 2 + \left\lfloor \log_2 \left(\frac{r-l}{2} \right) \right\rfloor \\
&= \left\lfloor \log_2 2 + \log_2 \left(\frac{r-l}{2} \right) \right\rfloor \\
&= \lfloor \log_2 (r-l) \rfloor
\end{aligned}$$

Fall 2: $(r-l)$ ist ungerade.

Dann ist auch $r+l = (r-l) + 2l$ ungerade, und mit (IV) folgt weiter:

$$\begin{aligned}
h_l &= 1 + \left\lfloor \log_2 \left(\frac{r+l-1}{2} - l \right) \right\rfloor \\
&= \log_2 2 + \left\lfloor \log_2 \left(\frac{r-l-1}{2} \right) \right\rfloor \\
&= \left\lfloor \log_2 2 + \log_2 \left(\frac{r-l-1}{2} \right) \right\rfloor \\
&= \lfloor \log_2 (r-l-1) \rfloor
\end{aligned}$$

Wegen der Monotonie des Logarithmus gilt

$$\lfloor \log_2(r-l-1) \rfloor \leq \lfloor \log_2(r-l) \rfloor$$

Außerdem gilt $\lfloor \log_2(r-l-1) \rfloor < \lfloor \log_2(r-l) \rfloor$ gdw. $r-l=2^n$ für ein $n \in \mathbb{N}$. Da aber $(r-l)$ als ungerade und größer als 1 vorausgesetzt wurde, kann $(r-l)$ keine Zweierpotenz sein, und es gilt Gleichheit:

$$h_l = \lfloor \log_2(r-l-1) \rfloor = \lfloor \log_2(r-l) \rfloor$$

Also gilt in allen Fällen die Behauptung.

Damit ist eine untere Schranke von $\lfloor \log_2(r-l) \rfloor$ für die Höhe eines Blattes in $ST(l,r)$ gezeigt.

Also gilt für alle Blätter v von $ST(l,r)$:

$$h(v) \in \{\lfloor \log_2(l-r) \rfloor, \dots, \lceil \log_2(l-r) \rceil\} \subseteq \{\lfloor \log_2(l-r) \rfloor, \lfloor \log_2(l-r) \rfloor + 1\}$$

Hieraus folgt unmittelbar die zu zeigende Balanceeigenschaft von Segmentbäumen. □

(c) Sei ein Intervall $[b,e]$ gegeben, wobei $r-l>3$ und $l \leq b < e \leq r$ vorausgesetzt wird.

Zu zeigen:

Das Intervall $[b,e]$ wird beim Einfügen in $ST(l,r)$ in höchstens

$$S := \lceil \log_2(r-l) \rceil + \lfloor \log_2(r-l) \rfloor - 2$$

Standardintervalle zerlegt.

Hierzu zeigen wir zunächst die beiden folgenden Hilfssätze:

Lemma 1:

Sei $ST(l,r)$ ein Segmentbaum mit $r-l \geq 2$ und $[b,r]$ ein „rechtsbündiges" Intervall, $l \leq b < r$. Dann wird das Intervall $[b,r]$ beim Einfügen in $ST(l,r)$ in maximal $\lfloor \log_2(r-l) \rfloor$ Standardintervalle zerlegt.

Beweis von Lemma 1:

Aus den Teilaufgaben (a) und (b) wissen wir, daß die Höhe h_l des linkesten Blattes gleich $\lfloor \log_2(r-l) \rfloor$ und die aller anderen Blätter mindestens ebenso groß ist.

(IA) Sei $h_l = 1$. Dann muß $r-l=2$ oder $r-l=3$ gelten.

Sei zunächst $r-l=3$. Es gibt folgende Zerlegungen des rechtsbündigen Intervalles $[b,r]$ in Standardintervalle:

$$\begin{aligned} [b,r] &= [l,r] && \Rightarrow \quad 1 \text{ Standardintervall} \\ [b,r] &= [l+1,r] && \Rightarrow \quad 1 \text{ Standardintervall} \\ [b,r] &= [l+2,r] && \Rightarrow \quad 1 \text{ Standardintervall} \end{aligned}$$

Sei nun $r-l=2$. Damit ergeben sich folgende Möglichkeiten zur Zerlegung von $[b,r]$ in Standardintervalle:

$$[b,r] = [l,r] \quad \Rightarrow \quad 1 \text{ Standardintervall}$$
$$[b,r] = [l+1,r] \quad \Rightarrow \quad 1 \text{ Standardintervall}$$

Also benötigt man für $r-l=2$ und $r-l=3$ beim Einfügen eines rechtsbündigen Intervalles $[b,r]$ in den Segmentbaum $ST(l,r)$ maximal $\lfloor \log_2(r-l) \rfloor = 1$ Standardintervalle.

(IV) Für $r-l \in \{p | 2^{h_l} \leq p \leq 2^{h_l+1}-1\}$ sei bewiesen, daß beim Einfügen eines rechtsbündigen Intervalles $[b,r]$ in den Segmentbaum $ST(l,r)$ maximal $\lfloor \log_2(r-l) \rfloor =: h_l$ Standardintervalle alloziert werden.

(IS) $h_l \to h_l + 1$
Es gilt $2^{h_l+1} \leq r-l \leq 2^{h_l+2}$ gdw. $h_l + 1 = \lfloor \log_2(r-l) \rfloor$.

Fall 1: „Das aktuelle Standardintervall ist das einzufügende Intervall", also $b = l$. D.h. es genügt ein Standardintervall und es gilt $1 \leq h_l + 1$.

Fall 2: „Das einzufügende Intervall ist vollständig im rechten enthalten", also $b \geq \lfloor \frac{l+r}{2} \rfloor$. Es werden nach (IV) höchstens h_l Standardintervalle alloziert.

Fall 3: „Das einzufügende Intervall ist größer als das rechte", also $b < \lfloor \frac{l+r}{2} \rfloor$. Dann werden im rechten Teilbaum ein Standardintervall, im linken Teilbaum nach (IV) h_l Standardintervalle alloziert, insgesamt also $\lfloor \log_2(r-l) \rfloor$ viele.

Also gilt in allen Fällen die Behauptung, und die Induktion ist gezeigt.

$\square_{\text{Lemma 1}}$

Lemma 2:

Sei $ST(l,r)$ ein Segmentbaum mit $r-l \geq 2$ und $[l,e]$ ein „linksbündiges" Intervall, $l \leq e < r$. Dann wird das Intervall $[l,e]$ beim Einfügen in $ST(l,r)$ in maximal $\lceil \log_2(r-l) \rceil$ Standardintervalle zerlegt.

Beweis von Lemma 2:

Analog zum Beweis von Lemma 1. $\square_{\text{Lemma 2}}$

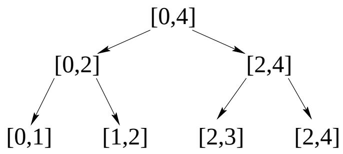
Nun wird unter Verwendung der beiden Lemmata die ursprüngliche Aussage gezeigt.

Beweis:

O.B.d.A. sei $l = 0$ und $r = n \in \mathbb{N}$. Durch vollständige Induktion über h wird nun gezeigt: Ist $ST(l,r)$ ein Segmentbaum mit $p := r-l$ und $2^{h-1}+1 \leq p \leq 2^h$, $h \in \mathbb{N}$, $h \geq 3$, oder $p = 4$, so ist die Anzahl der Standardintervalle, in die $[b,e]$ beim Einfügen zerlegt wird, höchstens gleich $S = \lceil \log_2(r-l) \rceil + \lfloor \log_2(r-l) \rfloor - 2$.

(IA) Für $h = 3$ und alle $p \in \{p | 2^{h-1}+1 \leq p \leq 2^h\} = \{5,6,7,8\}$ gilt die Behauptung, wie durch Aufzählen aller möglichen Fälle gezeigt werden kann.

Ebenfalls durch Aufzählen aller Fälle wird die Behauptung für $p = 4$ gezeigt:



Für die Intervalle $[0,1]$, $[0,2]$, $[0,4]$, $[1,2]$, $[2,3]$, $[2,4]$ und $[3,4]$ wird jeweils nur ein Standardintervall alloziert. Für $[0,3]$, $[1,3]$ und $[1,4]$ werden zwei Standardintervalle alloziert.

(IV) Sei die Behauptung für h schon gezeigt, also für alle $p \in \{x | 2^{h-1}+1 \leq x \leq 2^h\}$. Mit solchem p gilt für die Anzahl s der Standardintervalle, in die $[b,e]$ beim Einfügen zerlegt wird

$$s \leq S := \lceil \log_2(r-l) \rceil + \lfloor \log_2(r-l) \rfloor - 2$$

(IS) $h \to h+1$

Fall 1: „Das Intervall $[b,e]$ paßt vollständig in den linken Teilbaum", also $[b,e] \subseteq [l, \lfloor \frac{l+r}{2} \rfloor]$
Es werden also nur Standardintervalle des linken Teilbaumes alloziert, d.h. nach (IV) werden höchstens

$$\left\lceil \log_2 \left(\left\lfloor \frac{l+r}{2} \right\rfloor - l \right) \right\rceil + \left\lfloor \log_2 \left(\left\lfloor \frac{l+r}{2} \right\rfloor - l \right) \right\rfloor - 2 \leq S$$

Standardintervalle alloziert.

Fall 2: „Das Intervall $[b,e]$ paßt vollständig in den rechten Teilbaum"
Dieser Fall wird analog zu Fall 1 behandelt.

Fall 3: „Das Intervall $[b,e]$ überschneidet Bereiche beider Teilbäume", also $l \leq b < \lfloor \frac{l+r}{2} \rfloor < e \leq r$.
Ist $l = b$ und $r = e$, so wird nur ein Standardintervall (das der Wurzel) alloziert. Ansonsten sind die beiden Teilbäume zu betrachten.
Im linken Teilbaum $ST(l, \lfloor \frac{l+r}{2} \rfloor)$ werden s_l Standardintervalle für das rechtsbündige Intervall $[b, \lfloor \frac{l+r}{2} \rfloor]$ alloziert. Mit Lemma 1 sind dies höchstens

$$\begin{aligned} s_l &\leq \left\lfloor \log_2 (\lfloor \frac{l+r}{2} \rfloor - l) \right\rfloor \\ &= \left\lfloor \log_2 \left\lfloor \frac{r-l}{2} \right\rfloor \right\rfloor \\ &\overset{\text{siehe (b)}}{=} \left\lfloor \log_2 \frac{r-l}{2} \right\rfloor \\ &= \lfloor \log_2(r-l) \rfloor - 1 \end{aligned}$$

Im rechten Teilbaum $ST(\lfloor \frac{l+r}{2} \rfloor, r)$ werden s_r Standardintervalle für das linksbündige Intervall $[\lfloor \frac{l+r}{2} \rfloor, e]$ alloziert. Mit Lemma 2 sind dies höchstens

$$\begin{aligned} s_r &\leq \left\lceil \log_2(r - \lfloor \frac{l+r}{2} \rfloor) \right\rceil \\ &= \left\lceil \log_2 \left\lceil \frac{r-l}{2} \right\rceil \right\rceil \\ &\overset{\text{siehe (a)}}{=} \left\lceil \log_2 \frac{r-l}{2} \right\rceil \\ &= \lceil \log_2(r-l) \rceil - 1 \end{aligned}$$

Im linken und rechten Teilbaum werden zusammen also höchsten S viele Standardintervalle alloziert.

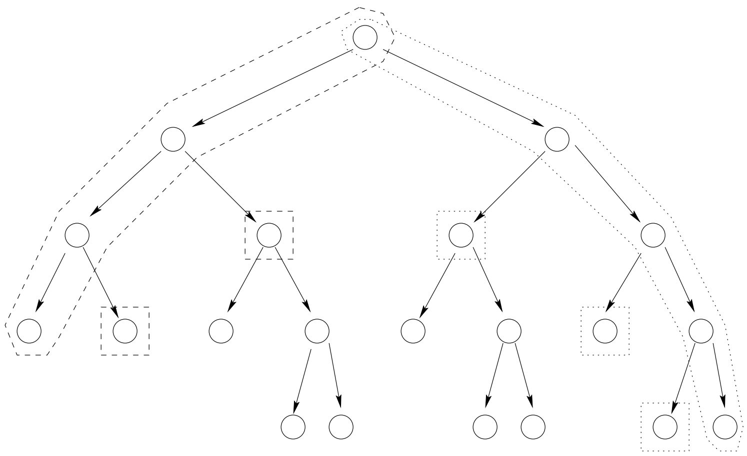
Also gilt in jedem Falle, daß höchstens $\lfloor \log_2(r-l) \rfloor + \lceil \log_2(r-l) \rceil - 2$ Standardintervalle alloziert werden. □

(d) **Zu zeigen:**

Es gibt ein Intervall $[b, e]$, das beim Einfügen in einen Segmentbaum $ST(l, r)$, $r - l > 3$, in $S := \lceil \log_2(r-l) \rceil + \lfloor \log_2(r-l) \rfloor - 2$ Standardintervalle zerlegt wird.

Beweis:

Wähle $[b, e] = [l+1, r-1]$. Es gilt $e - b = (r-1) - (l+1) = r - l - 2 > 1$, d.h. $[b, e]$ ist eine zulässige Wahl.



Auf dem Pfad von der Wurzel zum linkesten Blatt wird das Intervall $[b, e]$ für den rechten Sohn jedes inneren Knotens des Pfades alloziert. Dies ergibt $\lfloor \log_2(r-l) \rfloor - 1$ Standardintervalle (siehe Teilaufgabe (b)).

Analog wird auf dem Pfad von der Wurzel zum rechtesten Blatt das Intervall $[b, e]$ für den linken Sohn jedes inneren Knotens des Pfades alloziert. Dies ergibt $\lceil \log_2(r-l) \rceil - 1$ Standardintervalle (siehe Teilaufgaben (a) und (b)).

Insgesamt wird $[b, e] = [l+1, r-1]$ also in $S = \lceil \log_2(r-l) \rceil + \lfloor \log_2(r-l) \rfloor - 2$ Standardintervalle zerlegt. □