

Prof. Dr. T. Härder  
 Fachbereich Informatik  
 Arbeitsgruppe Datenbanken und Informationssysteme  
 Universität Kaiserslautern

## **Übungsblatt 4 – Lösungsvorschläge**

für die freiwillige Übung

**Unterlagen zur Vorlesung:** „[www.dvs.informatik.uni-kl.de/courses/DBSREAL/](http://www.dvs.informatik.uni-kl.de/courses/DBSREAL/)“

### **Aufgabe 1: *Pointer-Swizzling* bei Bäumen**

Gegeben ist ein vollständig belegter B\*-Baum der Klasse  $T(k, k^*, h)$ , der in seinen Blattseiten auf Sätze verweist, die keine weiteren Verweise enthalten.

Die B\*-Baum-Seiten besitzen folgendes Format:

Innerer Knoten	$Z_0$	$R_1$	$Z_1$	$R_2$	$Z_2$	$\dots$	$R_p$	$Z_p$	freier Platz
----------------	-------	-------	-------	-------	-------	---------	-------	-------	--------------

$R_i$  = Referenzschlüssel (Wegweiser),  $k \leq p \leq 2k$

$Z_i$  = Verweis auf Sohnseite

Blattknoten	P	$S_1$	$D_1$	$S_2$	$D_2$	$\dots$	$S_j$	$D_j$	freier Platz	N
-------------	---	-------	-------	-------	-------	---------	-------	-------	--------------	---

P = PRIOR-Zeiger, N = NEXT-Zeiger,  $k^* \leq j \leq 2k^*$

$S_i$  = Satzschlüssel

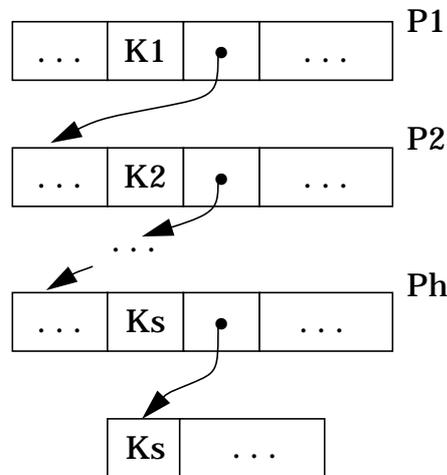
$D_i$  = Verweis auf Satz (referenzierte Speicherung)

Eine innere Seite (inkl. der Wurzel) habe also bei voller Belegung  $2k+1$  Verweise auf Seiten und eine Blattseite auf der Stufe  $h$  habe  $2k^*$  Verweise auf Sätze.

Anfangs seien keine Seiten des B\*-Baumes und keine der referenzierten Sätze im Hauptspeicher. Im folgenden soll Copy-Swizzling analysiert werden.

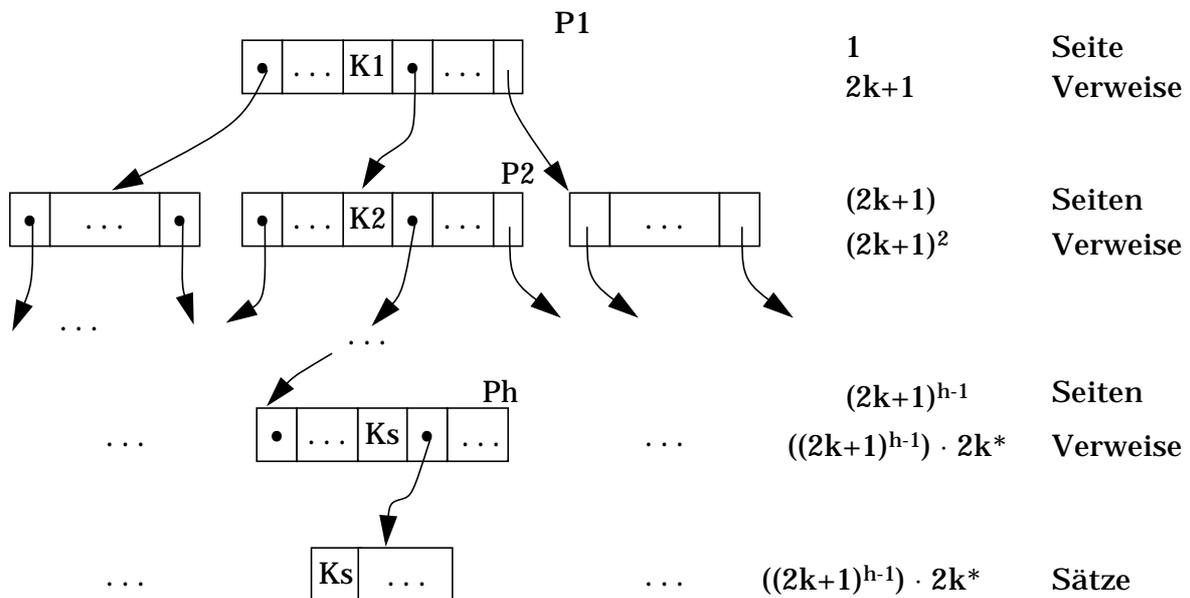
Skizzieren Sie für die verschiedenen Swizzling-Varianten die einzelnen Aktionen, die aus einem Suchvorgang für den Satz mit Schlüssel  $K_s$  resultieren, wenn dabei die Wurzelseite  $P_1$ , auf Baumebene 2 die Seite  $P_2$ , ... und auf Baumebene  $h$  die Seite  $P_h$  aufgesucht werden. Der Suchpfad ergebe sich durch die Schlüssel (Wegwei-

ser) K1 in P1, K2 in P2, ..., Ks in Ph:



Wie viele Seiten, Deskriptoren und Sätze sind jeweils einzulagern und aufzusuchen, wenn als Verfahren

a) **eager, direct**



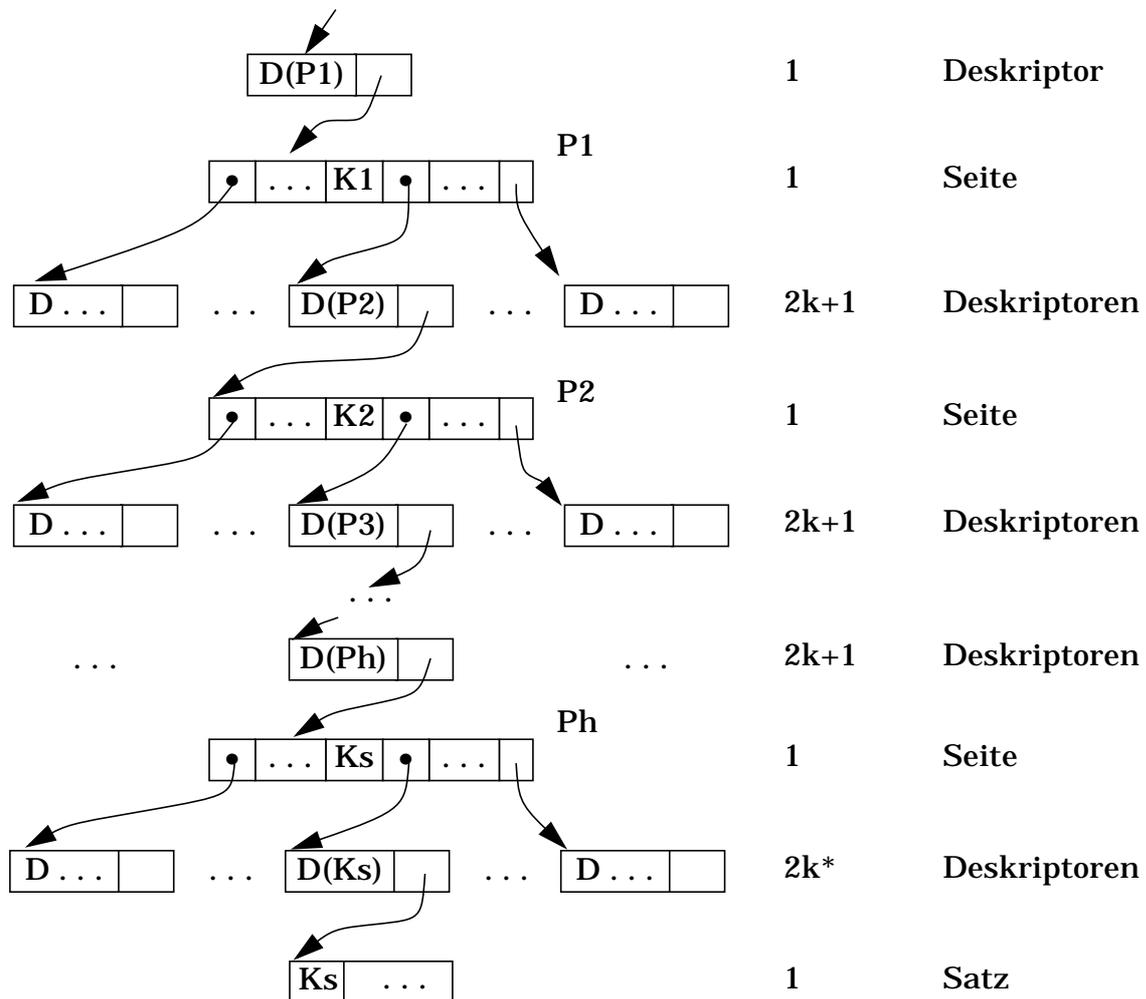
Einlagerung bei erstmaliger Referenz von P1:

- $(1 + \sum_{i=2}^h (2k+1)^{i-1})$     Seiten
- $((2k+1)^{h-1}) \cdot 2k^*$     Sätze

Lesen von

- h    Seiten
- 1    Satz

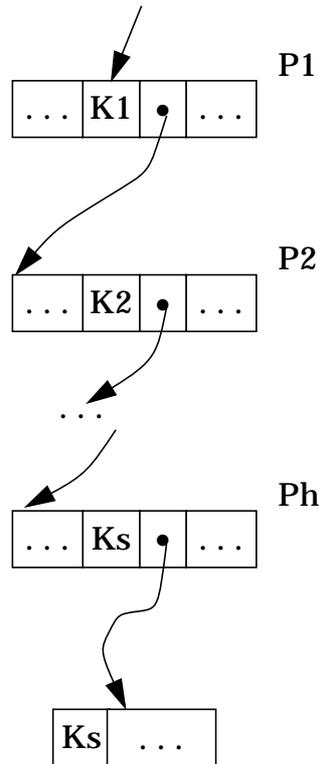
b) **eager, indirect**



- Beginn der Suche nach Ks:
  - Anlegen von D(P1)
- Zugriff auf P1
  - Einlagerung von P1
  - Anlegen von D(P2) und weiteren 2k Desk.
  - Suche nach K1
- Zugriff auf P2
  - ...
- Zugriff auf Ph
  - Einlagerung von Ph
  - Anlegen von D(Ks) und weiteren (2k\*-1) Desk.
  - Suche nach Ks
- Zugriff auf Ks
  - Einlagerung von Satz mit Ks als Schlüssel
  - Zugriff auf Satz Ks

- Insgesamt:**
- Einlagerung/Anlegen von
    - h Seiten
    - $1 + (h-1) \cdot (2k+1) + 2k^h$  Desk.
    - 1 Satz
  - Lesen von
    - h Seiten
    - h + 1 Deskriptoren
    - 1 Satz

c) **lazy, direct**

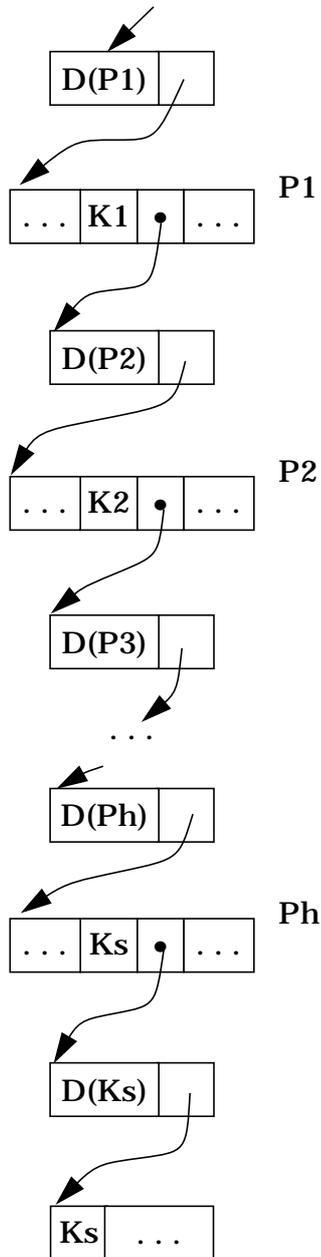


- Beginn der Suche nach Ks:
- Zugriff auf P1
  - Einlagern von P1
  - Suche nach K1
- Zugriff auf P2
  - ...
- Zugriff auf Ph
  - Einlagern von Ph
  - Suche nach Ks
- Zugriff auf Ks
  - Einlagern von Satz mit Ks als Schlüssel
  - Zugriff auf Satz Ks

**Insgesamt:**

- Einlagerung von
  - h Seiten
  - 1 Satz
- Lesen von
  - h Seiten
  - 1 Satz

d) **lazy, indirect**



- Beginn der Suche nach Ks:
  - Anlegen von D(P1)
- Zugriff auf P1
  - Einlagerung von P1
  - Suche nach K1
  - Anlegen von D(P2)
- Zugriff auf P2
  - ...
- Zugriff auf Ph
  - Einlagerung von Ph
  - Suche von Ks
  - Anlegen von D(Ks)
- Zugriff von Ks
  - Einlagerung von Ks
  - Lesen von Ks

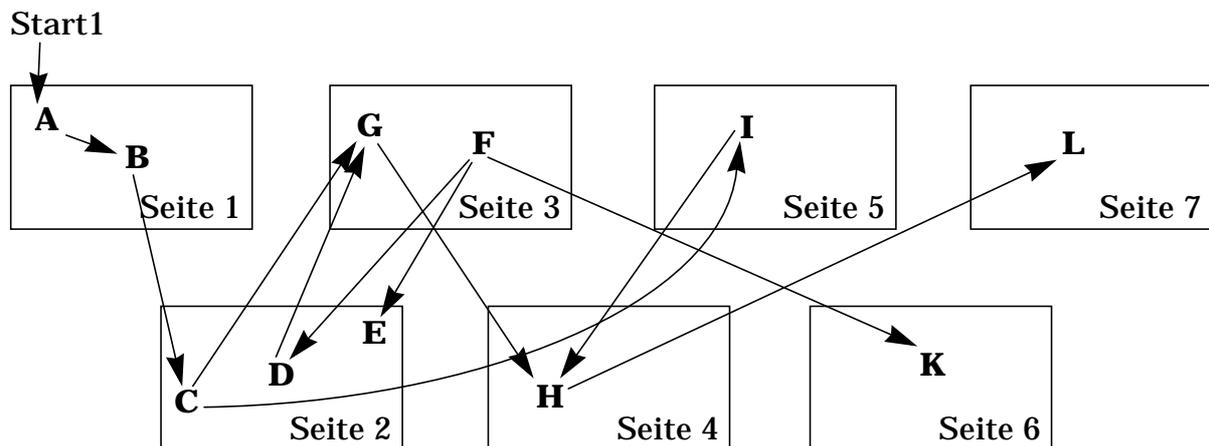
**Insgesamt:**

- Einlagern / Anlegen von
  - h Seiten
  - h + 1 Deskriptoren
  - 1 Satz
- Lesen von
  - h Seiten
  - h + 1 Seiten
  - 1 Satz

**Aufgabe 2: Pointer-Swizzling (direct in-place)**

187

Gegeben seien die Objekte A bis L, die auf externem Speicher in den Seiten 1 bis 7 wie unten abgebildet abgelegt sind. Die Pfeile repräsentieren dabei die Beziehungen der Objekte untereinander.



Es wird angenommen, daß diese Verweise als Externspeicheradressen vorliegen. Ein Transaktion referenziere die einzelnen Objekte in folgender Reihenfolge:

A B C G F E I H L K

Im folgenden werde immer von einem *Swizzling* mit den Eigenschaften *in-place*, *direct*, *lazy* ausgegangen, wobei bei der ersten Referenz auf eine Hauptspeicheradresse umgestellt wird.

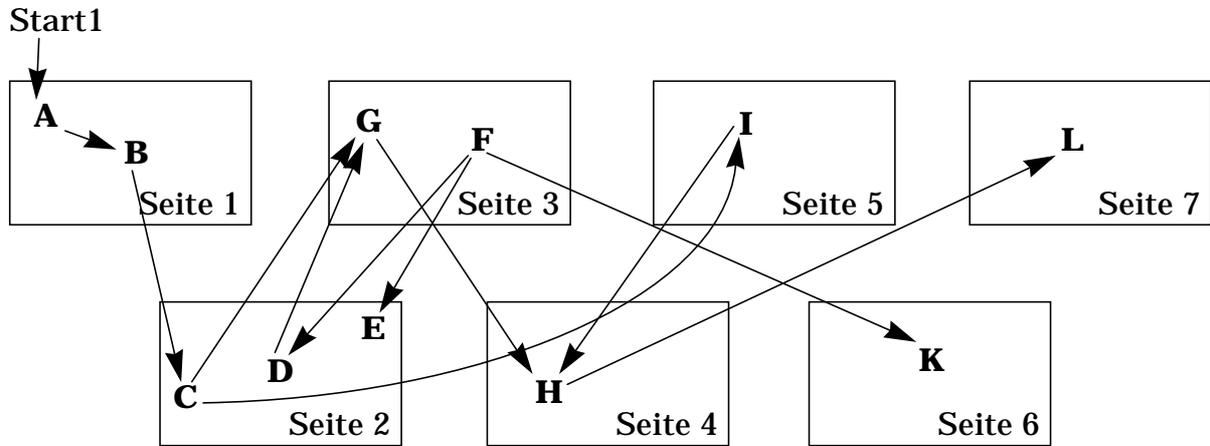
a) Welche Auswirkungen haben die obigen Objektreferenzen auf die Hauptspeicherbelegung, wenn davon ausgegangen werden kann, daß alle Seiten in den Hauptspeicher passen?

Bei Referenz eines Objektes wird die zugehörige Seite eingelagert. Die von einem Objekt ausgehenden Verweise werden nach dem lazy-Verfahren (bei erster Referenz (Nutzung des Verweises)) umgestellt.

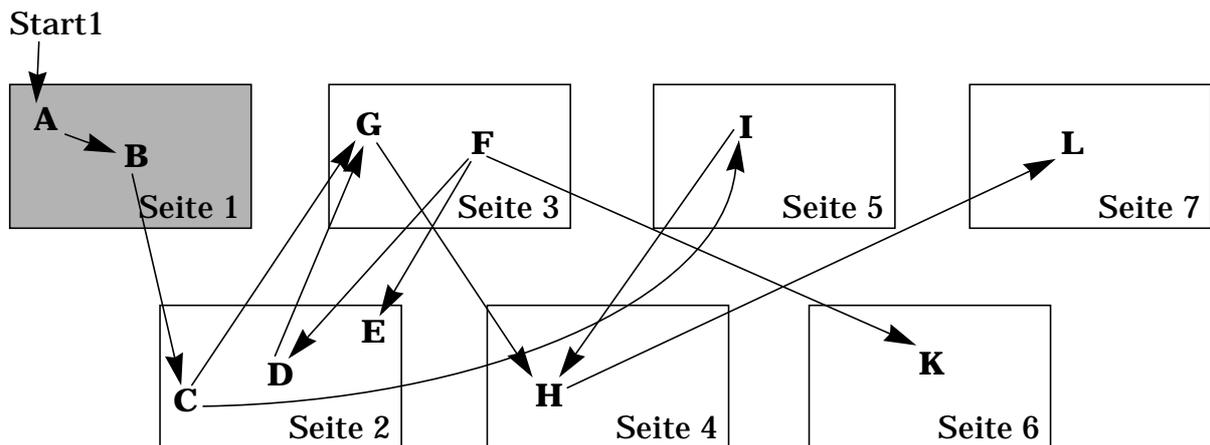
b) Angenommen es stünden nur 5 Seiten im Hauptspeicher zur Verfügung. Beim Einlesen der Seite mit Objekt L in Seite 7 müßte dann Platz geschaffen werden. Welche Seite(n) könnte(n) verdrängt werden?

Dunkelgraue Seiten sind eingelagert:

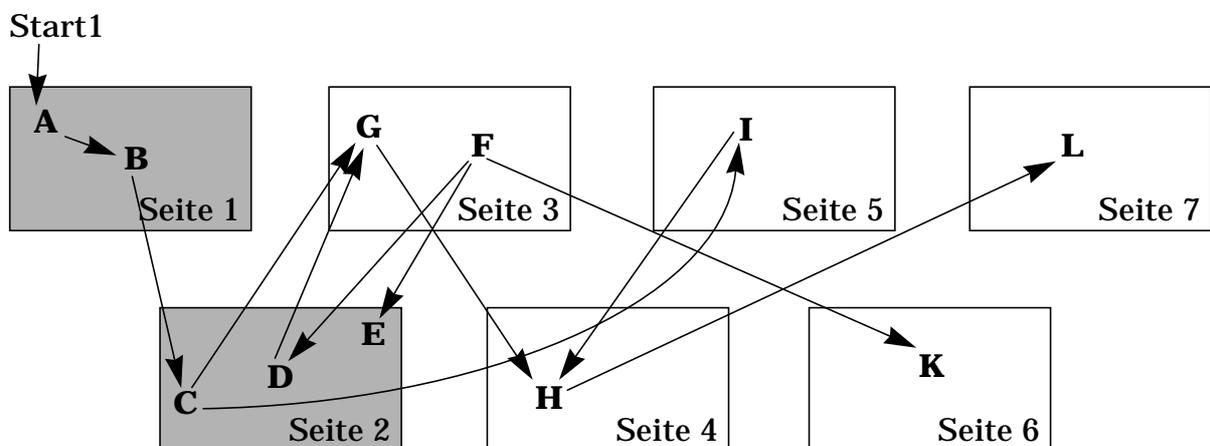
Ausgangssituation:



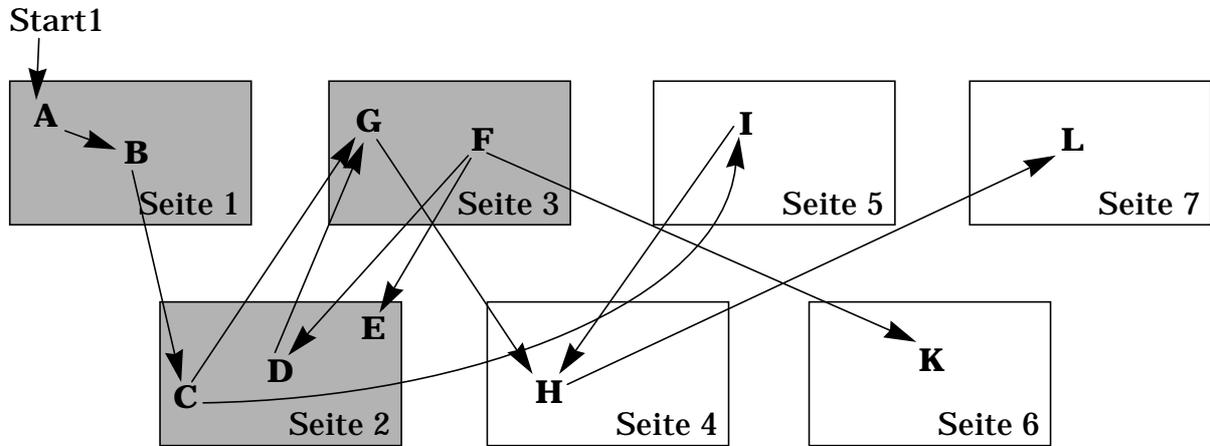
Nach der Referenz auf A und auf B:



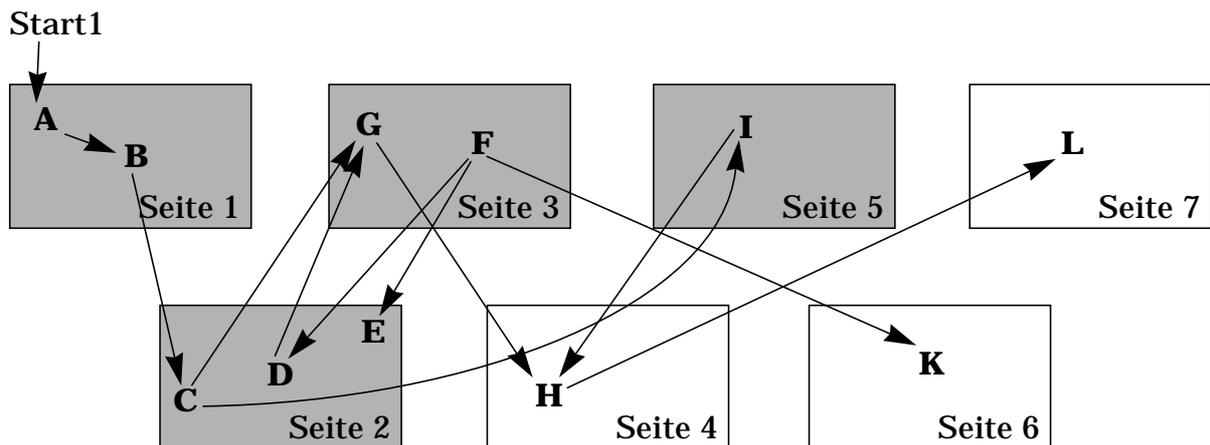
nach der Referenz auf C:



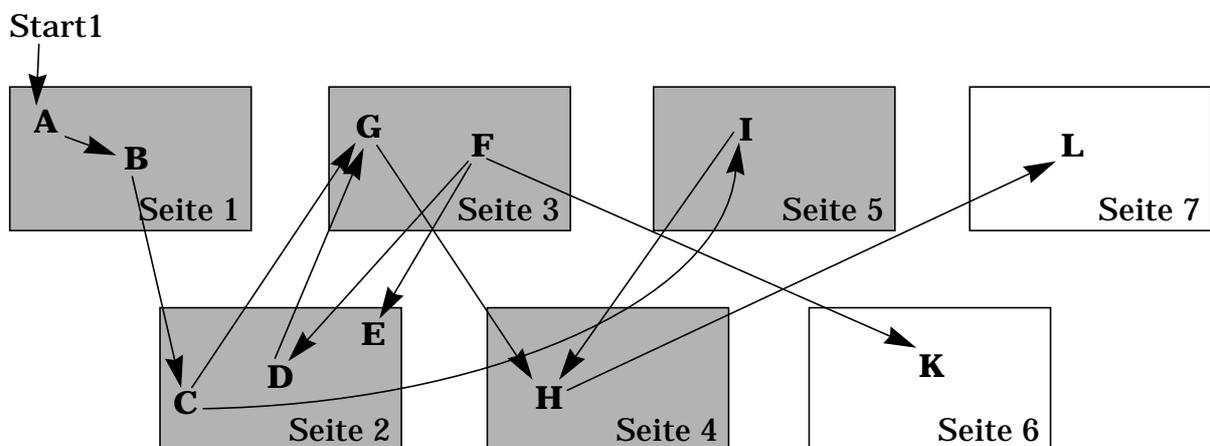
nach der Referenz auf G:



nach der Referenz auf F, E und I:



nach der Referenz auf H:



Da es keine Referenzen auf Objekte in Seite 1 gibt, könnte diese Seite verdrängt werden. Alle anderen Seiten können nicht so einfach verdrängt werden, weil dann Verweise auf in diesen Seiten enthaltene Objekte nicht mehr gültig sein können.

c) Wie wäre die Situation, wenn zwischenzeitlich in Seite 1 ein neues Objekt eingefügt worden wäre?

Wenn zwischenzeitlich ein Objekt in Seite 1 eingefügt worden ist, hängt eine mögliche Seitenersetzung davon ab, von welchen Objekten dieses neue Objekt referenziert wird (s. o.).

d) Welche Informationen für die einzelnen Seiten/Objekte werden beim *Pointer-Swizzling* benötigt?

Informationen in einer Seite:

1. welche Objekte sind in der Seite
2. an welcher Stelle stehen diese Objekte innerhalb der Seite

Informationen über die Objekte:

1. welche externe Adresse wird auf welche interne Adresse abgebildet
2. welche Objekte werden von einem Objekt referenziert

### **Aufgabe 3: Abbildung von Datensätzen in Seiten**

Jede Datenseite benötigt gewisse Informationen, die der Identifikation und der Selbstbeschreibung dienen. Sie werden im sogenannten *Page-Header* abgelegt. Überlegen Sie, welche Informationen zur Beschreibung einer Datenseite notwendig sind.

a) Entwerfen Sie für den *Page-Header* eine geeignete Datenstruktur und legen Sie den Typ der einzelnen Elemente dieser Struktur fest.

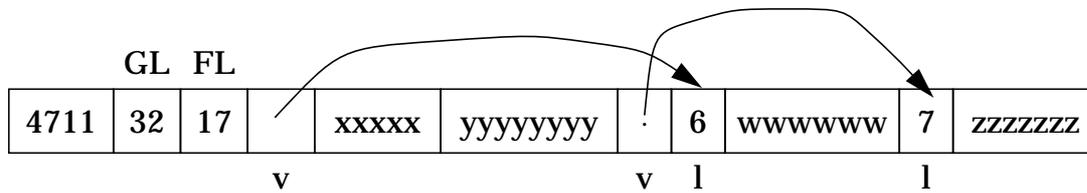
Seitenheader-Informationen:

- Segment\_Nr.,
- Seiten\_Nr.,
- next\_ptr zur sequentiellen Verarbeitung
- # freier Bytes
- Seitentyp (Datenseite, Freispeicherverwaltung, Zugriffspfad, ...)

Die Sätze einer Tabelle TAB seien nach dem 4. Verfahren im Skript als Speicherungsstrukturen abgelegt. Anfangs haben sie 4 Felder und sind im Katalog durch v, f5, f8, v beschrieben.

Ein bestimmter Satz mit SKZ = 4711 habe folgende Repräsentation, wobei der Zeiger eines v-Feldes 2 Byte und die Längendarstellung eines v-Feldes (l) 1 Byte aus-

machen.



Durch

Alter Table TAB (Add A-Feld var) und Alter Table TAB (Add B-Feld fixed (3))

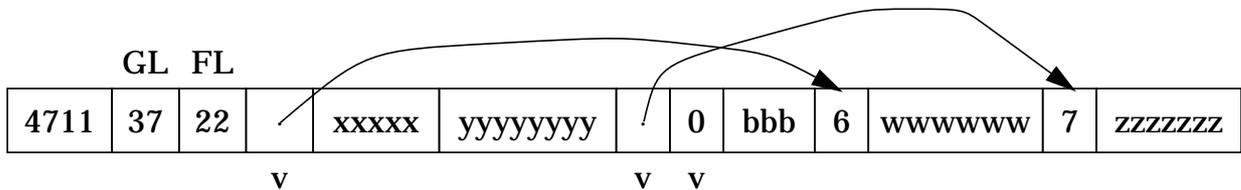
werden dynamisch zwei Felder an TAB-Sätze angehängt, was zunächst nur Auswirkungen auf die Katalog-Information von TAB hat: v, f5, f8, v, v, f3. Die einzelnen Sätze von TAB werden erst bei ihrer Aktualisierung mit Werten für A-Feld und B-Feld modifiziert.

Welche Änderungen an der Speicherungsstruktur von Satz 4711 ergeben sich, wenn

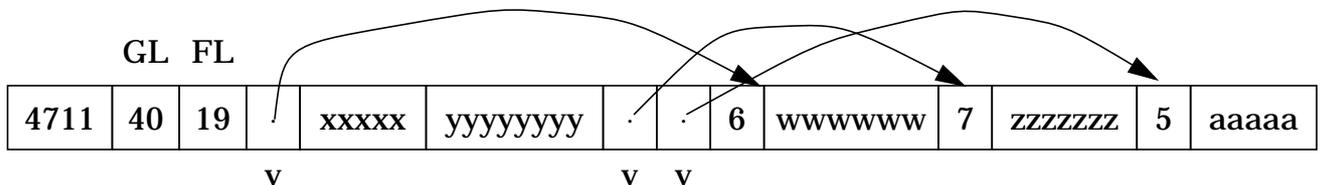
b) beide neuen Felder noch keinen Wert besitzen

keine Änderung

c) B-Feld den Wert bbb und A-Feld noch keinen Wert besitzt



d) A-Feld den Wert 5aaaaa und B-Feld noch keinen Wert besitzt



e) beide neuen Felder die obigen Werte haben?

