7.2 Interne Ebene: Hashorganisation (eindimensional, dynamisch)

- gegen erstes Problem bei statischem Hashen: Einsatz dynamischer Hashfunktionen

- Lineares Hashen (Litwin, 1980):
  - h_i: dom(Primärschlüssel) $\rightarrow$ {0,..., 2^i*N} Folge von Hashfunktionen, N Anfangsgröße des Hashverzeichnisses
  - Heuristik: es gilt
    $h_{i+1}(w) = h_i(w)$ für etwa die Hälfte aller w $\in$ dom(prim.)
    $h_{i+1}(w) = h_i(w) + 2^i*N$ für die andere Hälfte
    mit $h_i(w) = w \bmod 2^i*N$
    für ein w sind höchstens zwei Hashfunktionen zuständig, Entscheidung zwischen diesen beiden durch Parameter
    * sp: Splitpointer, gibt an, welche Seite als nächstes geteilt wird, initial. mit 0
    * lv: Level, gibt an, welche Hashfunktion benutzt wird, initial. mit 0
  - Lookup: geg.: Primärschlüsselwert w
    gesucht: Hashnummer s im Hashverzeichnis:

```
IF (sp==0) s = h_lv(w);
           ELSE { s = h_lv-1(w);
                  IF (s < sp) s = h_lv(w);
                  }
```

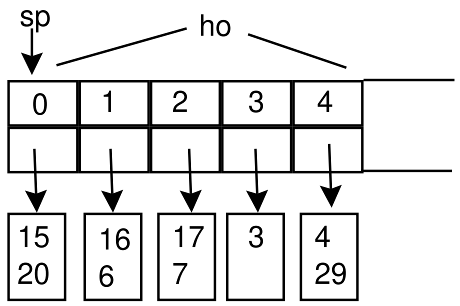
Hashorganisation (eindimensional, dynamisch)

- Einfügen, Löschen: wie Lookup,
  - falls Füllfaktor (Anzahl abzuspeichender Tupel/durchschn. Anzahl mögl. Sätze auf Seite) höher als vorgegebener Maximalwert automatisch (lokale) Reorganisation: Sätze, auf die sp zeigt, werden gemäß h_{lv} neu verteilt (d.h. etwa die Hälfte der Sätze wird auf Seiten unter Hashnr. 2^{lv-1}*N+sp geschoben),
sp = sp+1
IF (sp==2^{lv-1}*N) {lv = lv+1; sp = 0;}

    Ausnahme: erstes Splitten: lv = lv+1, sp=sp+1
  - falls Füllfaktor niedriger als vorgegebener Minimalwert, wird autom. lokal reorganisiert: zwei Hashnummern und ihre Seiten wieder zusammenlegen,
sp = sp -1
evtl. lv = lv -1

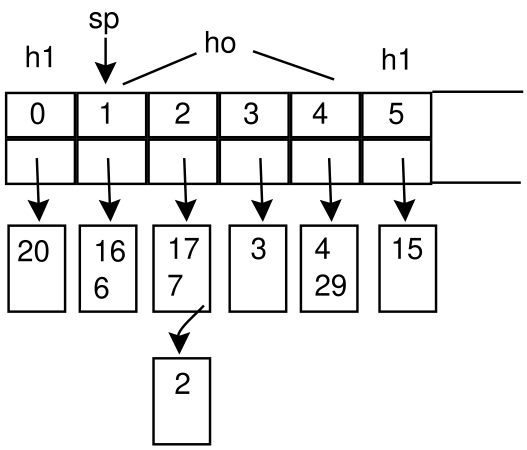
Hashorganisation (eindimensional, dynamisch)

Beispiel: $h_i(w) = w \bmod 2^i*5$ (N=5)



Ausgangszustand
lv=i=0

Annahme: beim Einfuegen von 2 wird Fuellfaktor ueberschritten



lv = i = 1

$h_0(w) = w \bmod 2^0*5$
$h_1(w) = w \bmod 2^1*5$

Quicky: was passiert bei Suche nach 15, 20, 3, 2?

Nachteil des lin. Hasing: evtl. werden Seiten gespittet, die gar nicht voll sind