Skip-Listen

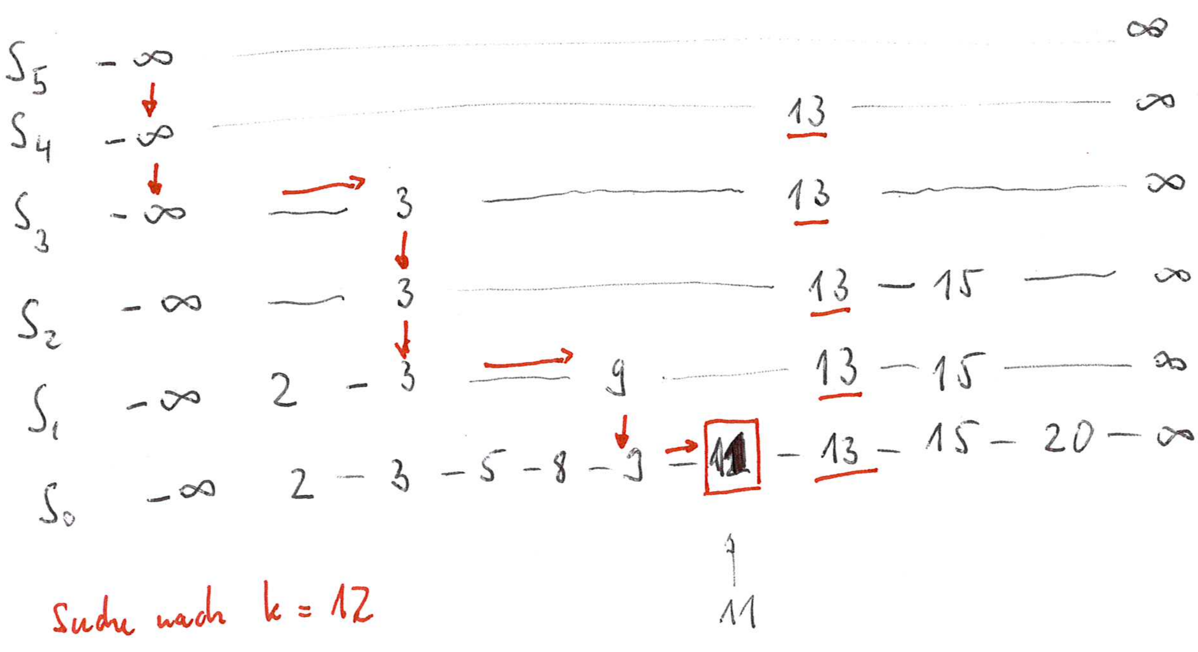
Skip-Listen sind eine weitere Datenstruktur für den ADT Dictionary. Zur Konstruktion wird ein Zufallsgenerator verwendet. Bei der Laufzeitanalyse untersucht man nicht den schlechtesten Fall, sondern den Erwartungswert (erwartete Laufzeit).

Eine Skip-Liste ist eine Folge von geordneten doppelt verketteten Listen $S_0, S_1, \dots, S_h$ wobei die Einträge von S_{i+1} immer eine Teilmenge der Einträge von S_i sind.

S_0 enthält alle Einträge der Datenstruktur.

Jede Liste hat zusätzlich zwei Dummy-Einträge mit den Schlüsseln $-\infty$ und ∞ (S_h hat nur $-\infty$ und ∞)

Darstellung:



Suche nach $k = 12$

Jede Position p hat Referenzen zu den Nachbarn links, rechts, über und unter sich (eventuell null)

before (p), after (p), above (p), below (p)

Die Suche nach einem Schlüssel k beginnt immer in der obersten Liste. Ausgegeben wird die Position in S_0 mit dem größten Schlüssel $\leq k$

```
SkipSearch (k)
p = Position von -∞ in S_h , h > 0
while (below(p) ≠ null)
        p = below(p)
        while  key(after(p)) ≤ k
                p = after(p)

return p
```

Beispiel: SkipSearch (12)

Einfügen von Eintrag mit Schlüssel k

1) SkipSearch (k) = p
2) Eintrag erfolgt in S_0 direkt hinter p
3) Höhe des "Turm" des neuen Eintrags wird durch eine Folge von Münzwürfen bestimmt:
   Ergebnis 1: $S_i \rightarrow S_{i+1}$
   Ergebnis 0: Stop
4) Für Aktualisierung der doppelt verketteten Listen die Positionen nutzen, an denen Abwärtsbewegungen erfolgt sind.

Löschen eines Eintrags mit Schlüssel k:

1) SkipSearch (k) = p

2) Wenn $k \neq key(p)$ NO_SUCH_KEY

Sonst p und alle Positionen darüber löschen und Listenstrukturen rekonstruieren.

Probabilistische Analyse

1) Höhe der Skip-Liste

Wahrscheinlichkeit, dass ein bestimmter Eintrag in S_i $(i>0)$ vorkommt ist $\frac{1}{2^i}$ Münzwurf $\underbrace{1, 1, \ldots, 1}_{i}$

Wahrscheinlichkeit, dass einer von n Einträgen in S_i vorkommt ist $\leq \frac{n}{2^i}$

↑ Ereignis A_i

$$Pr(h > i) = P(A_i) \leq \frac{n}{2^i}$$

$$Pr(h > 3 \log_2 n) \leq \frac{n}{2^{3\log_2 n}} = \frac{n}{n^3} = \frac{1}{n^2}$$

sehr klein!

Erwartungswert $\mathbb{E}(h) \in O(\log n)$

---

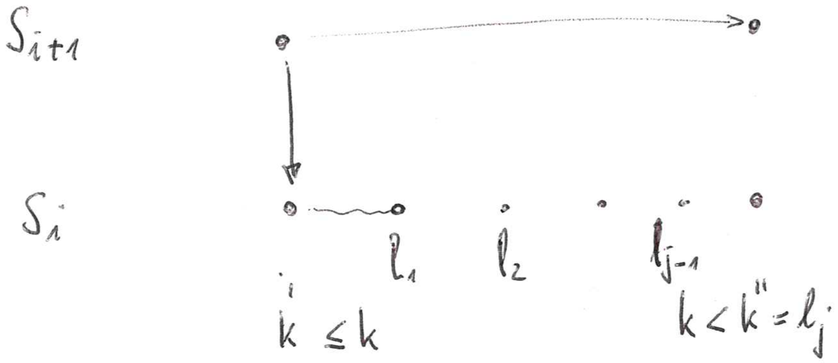
2) Anzahl der Positionen $\hat{=}$ Speicherbedarf

$$\sum_{i=0}^{h} \frac{n}{2^i} = n \sum_{i=0}^{h} \frac{1}{2^i} < 2n$$ <u>im Erwartungswert!</u>

3) Erwartete Laufzeit von SkipSearch(k)

- Es gibt $h \in O(\log n)$ Abwärtsbewegungen
- Zeigen, dass nach jedem Abwärtsschritt (von S_{i+1} nach S_i) die erwartete Anzahl von Schlüsselvergleichen in S_i höchstens 2 ist

Situation:



- Abwärts bei $k' \leq k$ in S_{i+1} weil Nachfolger $k'' > k$
- Seien $l_1, l_2, \ldots, l_j = k''$ die Nachfolger von k' in S_i

$\Rightarrow$ Beim Einfügen dieser Schlüssel gab es jeweils i mal eine 1 und dann bei $l_1, \ldots, l_{j-1}$ eine 0, aber bei l_j eine 1 (denn $k'' \in S_{i+1}$)

$\Rightarrow Pr(j=1) = \frac{1}{2}, \; Pr(j=2) = \frac{1}{4}, \ldots$

Anzahl der Schlüsselvergleiche T_i in S_i ist $\leq j$

$$E(T_i) \leq 1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + \ldots + n \cdot \frac{1}{2^n} \leq \sum_{j=1}^{\infty} j \cdot \frac{1}{2} \cdot \left(\frac{1}{2}\right)^{j-1} = 2$$

Erwartungswert für geometr. Vert. mit $p = \frac{1}{2}$

- Erwartete Laufzeit von SkipSearch$(k) \leq 3 \cdot h \in O(\log n)$

4) Folglich ist auch die erwartete Laufzeit von Einfüge- und Löschoperationen in $O(\log n)$