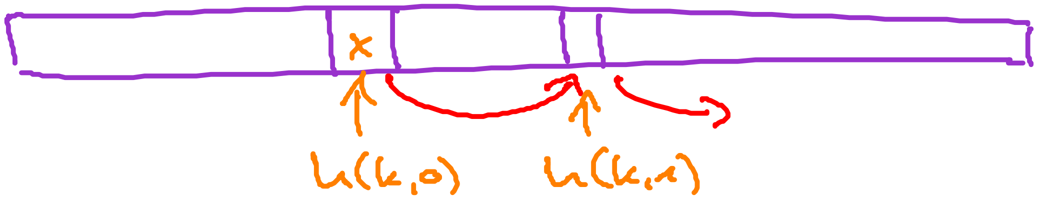
Kollisionsbehandlung beim Hashing:

2) open addressing

$h : U \times \{0, \dots, m-1\} \to \{0, \dots, m-1\}$

$h(k,i) \hat{=}$ Adresse nach i erfolglosen Versuchen beim Einfügen von k.



Analyse von open addressing

Mache hierzu „Gleichverteilungsannahme“: Die jeweils nächste Adresse wird zufällig und gleichverteilt aus $\{0,\dots,m-1\}$ gewählt.

Satz: Bei Auslastung $\alpha = \frac{n}{m} < 1$ ist

$$E(\#\text{Sondierungen beim Einfügen}) \leq \frac{1}{1-\alpha}.$$

Für Beweis betrachte Urnenmodell:

m Kugeln, w weiße, s schwarze $(m = w+s)$

ziehe gleichverteilt mit Zurücklegen

$$E(\#\text{Ziehungen bis weiße Kugel gezogen}) = \frac{m}{w}$$

Bsp.:

$$E(\#\text{Würfelwürfe bis 6 gewürfelt}) = \frac{6}{1} = 6$$

$$E(\#\ \text{„ „ 1 oder 2 „}) = \frac{6}{2} = 3$$

Begriffe aus der W'Theorie:

diskrete Zufallsgröße X (endlicher Fall) hat Werte $x_1, x_2, \dots, x_n$ die mit W'keiten $p_1, p_2, \dots, p_n$ auftreten.

Bsp. 1 Würfel

1	2	3	4	5	6
1/6	1/6	1/6	1/6	1/6	1/6

Würfeln mit 2 Würfeln, $X = \sum$ Augenzahlen

Werte	2	3	4	5	6	7	8	9	10	11	12
	1/36	2/36	3/36	-	-	-	-	-	-	2/36	1/36

Erwartungswert der Zufallsgröße X ist

$$E(X) = \sum_{i=1}^{n} x_i \cdot p_i$$

1 Würfel: 3,5
2 Würfel: 7

Analog für den abzählbar unendlichen Fall:

X hat Werte $x_1, x_2, x_3, \dots, x_i, \dots$
mit W'keiten $p_1, p_2, p_3, \dots, p_i, \dots$

$$E(X) = \sum_{i=1}^{\infty} x_i\, p_i \quad \text{falls Reihe absolut konverg.}$$

Anwendung auf das Urnenmodell.

Kugel ist weiß mit W'keit $\frac{w}{m} =: p$
" " schwarz " " $\frac{s}{m} = 1 - p =: q$

X = # Ziehungen bis weiße Kugel gezogen

Werte von X		W'keit dafür
1	w	$p_1 = p$
2	sw	$p_2 = q \cdot p$
3	ssw	$p_3 = q^2 \cdot p$
$\vdots$		
i	$\underbrace{s \dots s}_{i-1} w$	$p_i = q^{i-1} \cdot p$

$$\sum_{i=1}^{\infty} p_i = \sum_{i=1}^{\infty} q^{i-1} \cdot p = p \cdot \sum_{i=0}^{\infty} q^i = p \cdot \frac{1}{1-q} = \frac{p}{p} = 1$$

$$E(X) = \sum_{i=1}^{\infty} i \cdot q^{i-1} \cdot p = p \cdot \underbrace{\sum_{i=1}^{\infty} i \cdot q^{i-1}}_{S :=}$$

$$\begin{aligned}
S &= 1 \cdot q^0 + 2 \cdot q^1 + 3 \cdot q^2 + 4 q^3 + \dots \\
&= q^0 + q^1 + q^2 + q^3 + \dots \quad \} \, 1 \cdot \sum_{i=0}^{\infty} q^i \\
&\quad + q^1 + q^2 + q^3 + \dots \quad \} \, q \cdot \sum_{i=0}^{\infty} q^i \\
&\quad + q^2 + q^3 + \dots \quad \} \, q^2 \cdot \sum_{i=0}^{\infty} q^i \\
&\quad + q^3 + \dots \quad \} \, q^3 \cdot \sum_{i=0}^{\infty} q^i \\
&\quad \vdots \\
&= \left(\sum_{i=0}^{\infty} q^i \right) \cdot \left(\sum_{i=0}^{\infty} q^i \right) = \frac{1}{p} \cdot \frac{1}{p} = \frac{1}{p^2}
\end{aligned}$$

(beachte: Reihe ist absolut konvergent)

$$\Longrightarrow E(X) = \frac{1}{p} = \frac{m}{w}$$

Anwendung auf open addressing:

Sondierungen $\hat{=}$ Ziehen aus Urne

$E(\#\text{Ziehungen bis freier Platz gefunden})$

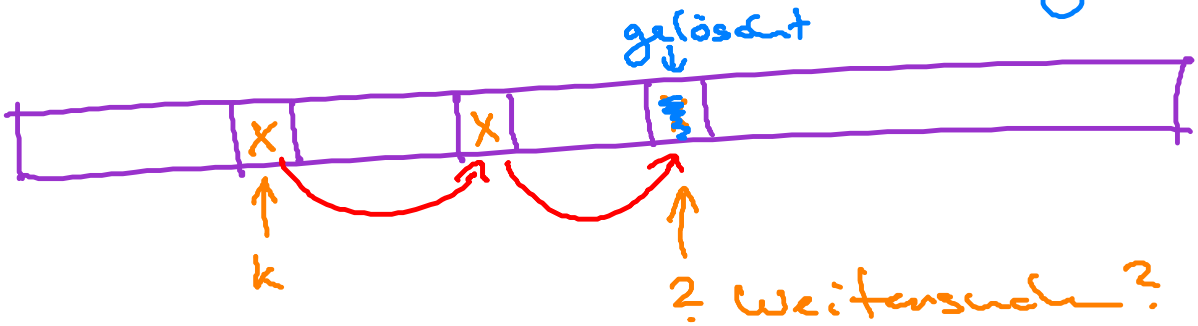
$$= \frac{m}{m-n} = \frac{1}{1-\frac{n}{m}} = \frac{1}{1-\alpha}$$

$\Rightarrow$ Satz. $\square$

Einfügen: im Mittel konstant (falls $\alpha < 1$ konstant)

Suchen: dieselbe Folge von Sondierungen wie beim Einfügen (sofern in der Einfügefolge nicht bereits Elemente gelöscht wurden).

Löschen bereitet daher Schwierigkeiten!



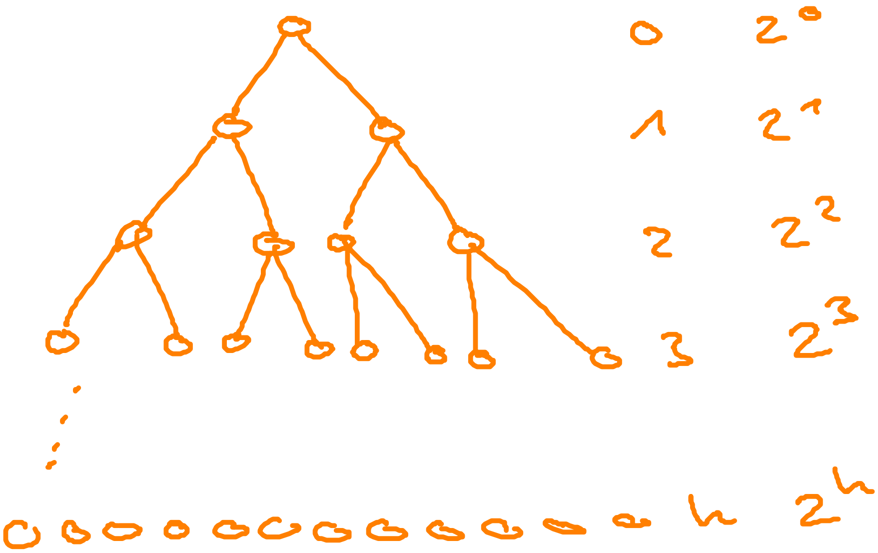
Daher: open addressing meistens beschränkt auf Hashing ohne Löschen.

Zusammenfassung:

Aufwand pro Operation $\mathcal{O}(1)$ im Mittel, aber im Worst Case $\Omega(n)$

Vergleich zu Suchbäumen:

- im Worst Case $O(\log n)$
- Mittlerer Aufwand $\Omega(\log n)$, denn:



Beobachtung: Mindestens die Hälfte aller Knoten liegen in Tiefe $\geq \log n$

$\Longrightarrow$ Average Case Aufwand für Suchen ist $\Omega(\log n)$.

Noch zwei weitere Aspekte von <u>Hashing</u>:

1) Universelles Hashing
2) Perfektes Hashing

(nicht im Skript, siehe daher CLRS (Cormen et al.))

1) Universelles Hashing:

Idee: Vermeide Worst Case. Wir stellen uns einen bösen Gegenspieler vor, der unser Hashing-Verfahren kennt und durch Auswahl spezieller Schlüssel in den Worst Case treibt.

→ Randomisieren zur Laufzeit des Verfahrens, d.h. wir wählen eine zufällige Hash-Fkt. Diese wird dann für gesamte Dauer des Verfahrens verwendet.

Def.: Eine Klasse H von Hash-Fkt. heißt universell bezüglich Größe m der Hash-Tabelle falls:

$\forall$ Schlüsselpaare $x, y \in U$ mit $x \neq y$ gilt:

$$|\{h \in H \mid h(x) = h(y)\}| \leq \frac{|H|}{m}$$

Idee: Wähle $h \in H$ zufällig und gleichverteilt, dann ist die W'keit für Kollision von x und y $\leq \frac{1}{m}$.

Frage: Existiert eine solche universelle Klasse? (→ später)

Satz: H sei universell für m, $h \in H$ zufällig und gleichverteilt gewählt. Werden nun $n \leq m$ Schlüssel mit h eingefügt und ist x ein fester Schlüssel, so gilt

$$E(\#\text{Kollisionen mit } x) < 1.$$

Beweis: Für Schlüsselpaar $y \neq z$ sei

$$C_{yz} = \begin{cases} 1 & \text{falls } h(y) = h(z) \\ 0 & \text{sonst} \end{cases} \quad \begin{matrix} \leftarrow \text{W'keit} \leq \frac{1}{m} \\ \leftarrow \quad " \quad \geq \frac{m-1}{m} \end{matrix}$$

$$E(C_{yz}) \leq \frac{1}{m}$$

Sei $C_x = \#$ Kollisionen mit x, d.h.

$$C_x = \sum_{y \neq x} C_{yx}$$

$$E(C_x) = E\left(\sum_{y \neq x} C_{yx}\right) = \sum_{y \neq x} E(C_{yx}) \leq \frac{n-1}{m} < 1 \qquad \square$$

Angewendet auf Hashing mit chaining:
Mittlere Listenlänge < 2.
$\Longrightarrow$ worst case tritt im Erwartungswert nicht auf, Absicherung gegen bösen Gegenspieler klappt.

Zeige noch: Es gibt universelle Klasse von Hashfunktionen für beliebiges m:

Wähle dazu zunächst Primzahl p mit $p > x$ für alle $x \in U$.

Für beliebiges $a \in \mathbb{Z}_p^* = \{1, \dots, p-1\}$ und $b \in \mathbb{Z}_p = \{0, 1, \dots, p-1\}$ sei Hash-Fkt. $h_{a,b}$ wie folgt definiert:

$$h_{a,b}(x) = ((a \cdot x + b) \bmod p) \bmod m$$

Es sei $H = \{h_{a,b} \mid a \in \mathbb{Z}_p^*, b \in \mathbb{Z}_p\}$

<u>Bsp:</u> $p = 17$, $m = 6$, $a = 3$, $b = 4$

$$h_{3,4}(8) = (\underbrace{(3 \cdot 8 + 4) \bmod 17}_{11}) \bmod 6 = 5$$

<u>Satz:</u> H ist universell bezüglich m.