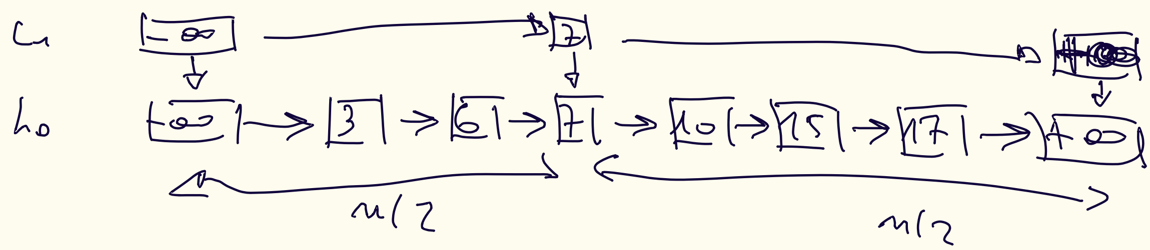
SKIP LIST

- alternative to BST
- supports operations in $O(\log_2 n)$ time

- BADS:
  - search operation: scan it entirely

- GOODS:
  - insertion } managed by pointers
  - deletion

- idea: - start from a list of items L_0 (with logical marks
  - create levels with vertical pointers
  (split L_0 in two and promote the middle element



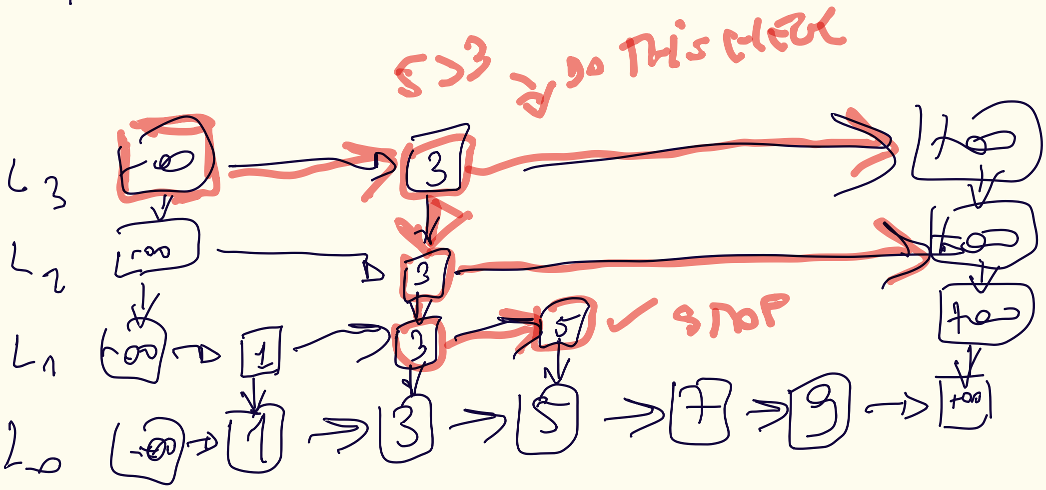
- Search(17): scanning L_1 allows to

- Given items, toss the coin:

How to build the skip list

1	H	T		
3	H	H	H	T
5	H	T		
7	T			
9	T			

search for $\boxed{5}$



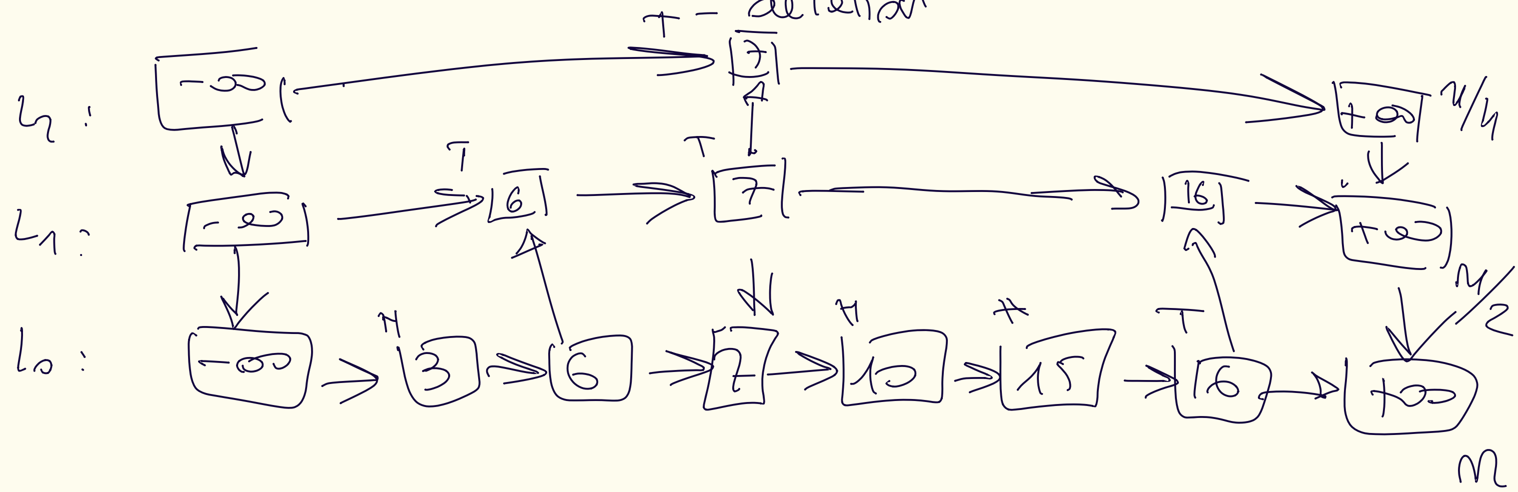
SKIP LIST

• key: on avg (and with high probability)

cost of: # keys

- insertion
- search $\Rightarrow O(\log_2 n)$
- deletion

• stop when I've got 1 item



• coin toss:

- TAIL $\Rightarrow$ promote

• search: vertical pointers (height)
orizontal pointers

• on AVG in L_1 I've: $\frac{1}{2} \cdot n = \frac{n}{2}$

• in L2 $\frac{1}{2} \cdot \frac{1}{2} \cdot n = \frac{n}{4}$

• we have a logaritmic # level

• (L) = # levels of skip list on n items $= \max_k L(k)$

$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad \|$ level of k-th item

• height of SL = maximum level (1)

→ level of 1 item key (Prob 1 specific element has at least l-levels)

• $\mathbb{P}(L(k) \geq l) = \left(\frac{1}{2}\right)^l$

(1) • when entire skip list has at least l-levels — probability level of skip list $\geq l$

(P) • $\mathbb{P}(L \geq l) = \mathbb{P}(\max_k L(k) \geq l) = \mathbb{P}(\exists k : L(k) \geq l) \leq$

of vertical keys → k

UNION BOUND

$$\leq n \cdot \mathbb{P}(k : L(k) \geq \ell) = n\left(\frac{1}{2}\right)^{\ell} = \frac{n}{2^{\ell}} \quad (*)$$

CASE

1) $\ell \leq \log_2 n \Rightarrow \frac{n}{2^{\ell}} \geq \frac{n}{2^{\log_2 n}} = \frac{n}{n} = 1$ NOT MEANINGFUL (For sure $\mathbb{P} \leq 1$)

2) $\ell > \log_2 n \Rightarrow \frac{n}{2^{\ell}} < \frac{n}{2^{\log_2 n}} = \frac{n}{n} = 1$ ✓

& WANT ESTIMATE: level of skip list > constant $\cdot \log_2 n$

=> PROB skip list is very high

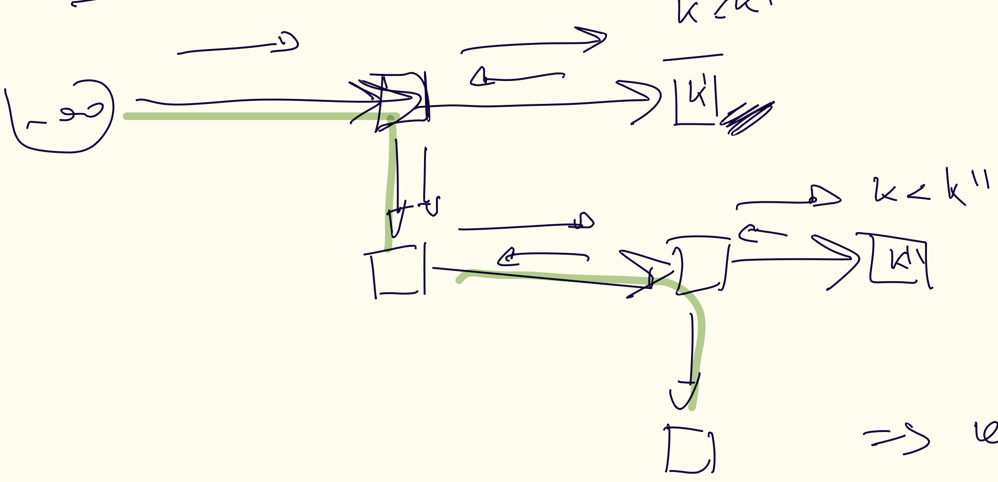
$$\mathbb{P}(L \geq \underbrace{c \cdot \log_2 n}_{\ell}) \leq \frac{n}{2^{c \cdot \log_2 n}} = \frac{n}{2^{(\log_2 n)^c}} = \frac{n}{n^c} = \frac{1}{n^{c-1}}$$

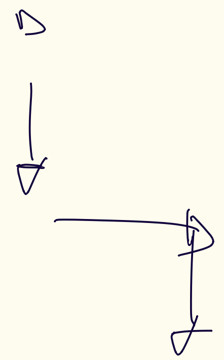
$(*)$ case 2

→ with high prob we have $\log_2 n$ vertical pointers

orizontal ~~vertical~~ pointers

search(k)

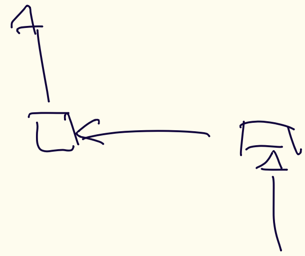




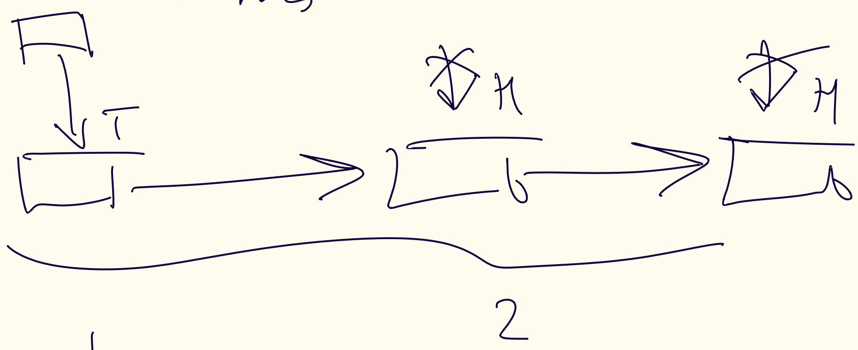
we know

=> vertical steps # $O(\log_2 n)$ w.h.p

looking back => I go up, I go back . . .



=> I go up just if there's a right pointer, so how many are orizontal pointers on AVG



=> P(2 head on head

[ON AVG (AT Every level) of head before get a ~~head~~ head (go up) I need 2 coin toss)

=> $\# \uparrow = \log_2 n$

$\# \rightarrow = \log_2 n$

=> Search cost $O(\log_2 n$

Dim search all skip list is $O(\log n)$

L: level of skip list

- VERTICAL POINTERS

$$\mathbb{P}(L(k) \geq \ell) = \left(\frac{1}{2}\right)^{\ell}$$

$$\mathbb{P}(L \geq \ell) = \mathbb{P}(\max_k L(k) \geq \ell) = \mathbb{P}(\exists k : L(k) \geq \ell) \leq$$

$$\leq n \cdot \mathbb{P}(L(k) \geq \ell) = n \cdot \frac{1}{2^{\ell}} = \frac{n}{2^{\ell}}$$

want logaritmic # of level

if $\ell \leq \log n \Rightarrow \frac{n}{2^{\ell}} \geq \frac{n}{2^{\log n}}$