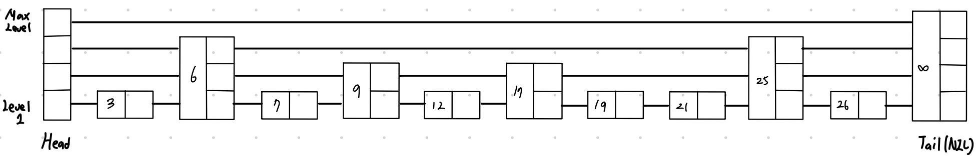
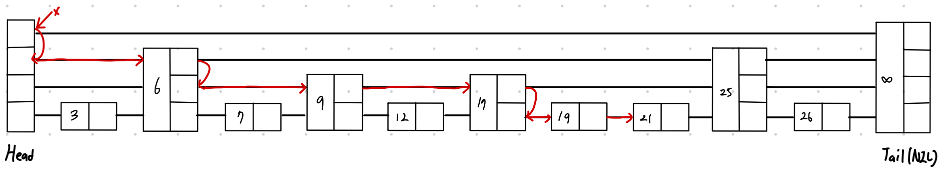
Skip Lists

- Operations: 1) Insert, 2) Delete, 3) Search
- Key Idea: Store keys in a sorted linked list, but with a twist
  1) Store extra pointers so that portions of the list can be skipped over while searching
  2) Extra pointers added probabilistically

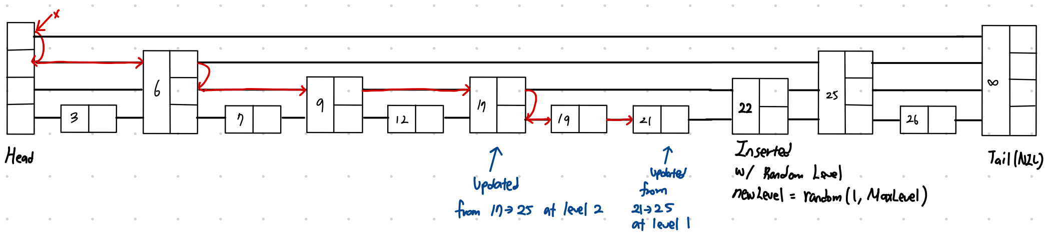
Level Rule: 1) A fraction p of nodes with level i pointers also have level i+1 pointers
2) Levels determined probabilistically.



Search (21):



Insert (22)



Insert
- Key idea: Search, then splice in
- Observation: Must keep track of all the nodes that need updating

Delete
- Key idea: Search, then splice out
- Observation: Must keep track of all the nodes that need updating

What's the P[arbitrary node has a pointer at level k]?

☆ An arbitrary node x

has a level 1 pointer with probability 1

has a level 2 pointer with probability p

has a level 3 pointer with probability p^2

...

has a level K pointer with probability p^{k-1}

Thus, P[arbitrary node has a pointer at level k] $= p^{k-1}$

Definition of $L(n)$: Expected level of node n.

$L(n) = \log_{\frac{1}{p}} n = -\log_p n$

· Significance of $L(n)$: Expected number of nodes with a pointer at level $L(n)$ is $\frac{1}{p}$. Why?

Proof:

E[number of nodes with pointer at level $L(n)$] $= n \cdot P$[arbitrary node has a pointer at level $L(n)$] $= n \cdot p^{L(n)-1}$

Definition of $L(n)$

$$p^{L(n)-1} = \frac{1}{p} \cdot p^{L(n)} = \frac{1}{p} \cdot p^{\log_{1/p} n}$$
$$= \frac{1}{p} \cdot p^{-\log_p n}$$
$$= \frac{1}{p} \cdot \frac{1}{p^{\log_p n}}$$
$$= \frac{1}{p} \cdot \frac{1}{n}$$

Important log rule here.

① $\log_x a^{\log_x b} = \log_x b \cdot \log_x a$

② $\log_{\frac{1}{x}} a = -\log_x a$

③ $a^{\log_a x} = x$

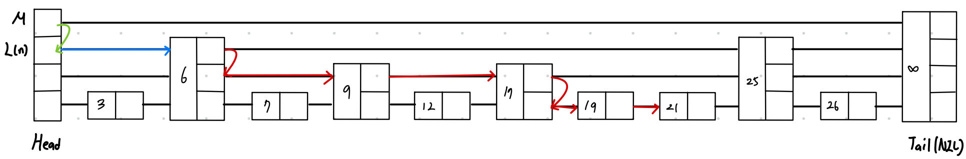
Hence, E[number of nodes with pointer at level $L(n)$] $= n \frac{1}{p} \frac{1}{n} = \frac{1}{p}$

Bounding the Search Path

: Expected number of hops to reach level $L(n)$ (A)

\+ Expected number of hops to the left at level $L(n)$ (B)

\+ Expected difference between maximum level at the time of search ($\equiv M$) and level $L(n)$, i.e., $E[M - L(n)]$ (C)



(A) Expected number of hops to reach level $L(n)$

Let $C(k)$ = expected length of a search path that climbs up K levels in an infinite list.

When traversing search path in the correct direction (left to right)

- move down w/ p
- move right w/ 1-p

In the reverse direction (right to left)

- move up w/ p
- move left w/ 1-p

Then, $C(0)=0$, since there's no level 0.

$C(k) = (1-p)(1+C(k)) + p(1+C(k-1))$

$(1-p)$: Probability of moving left

p: Probability of going up

$C(k-1)$: 1 level up, hence $C(k-1)$

$(1+C(k))$: Since it moves left (not going up to reach level $L(n)$), $C(k)$: expected # of hops remains same. But 1 steps happened.

$C(k) = 1 + C(k) - p - p\,C(k) + p + p.C(k-1)$

$C(k) = 1 + C(k) - p.C(k) + p.C(k-1)$

$p\cdot C(k) = 1 + p.C(k-1)$

$C(k) = \frac{1}{p} + C(k-1) = \frac{k}{p}$

Since $k = L(n)$, $A \le C(L(n)-1) = \frac{L(n)-1}{p} = \frac{1}{p}\cdot -\log_p n - 1$

(b) Expected number of hops to the left at level $L(n)$

: $E[\#$ of nodes w/ pointer at level $L(n)] = \frac{1}{p}$

(c) Expected difference between maximum level at the time of search ($=M$) and level $L(n)$, i.e., $E[M - L(n)]$

$P[M \ge L(n)+i]$: Probability that some node has a pointer at level $L(n)+i$

$P[M \ge L(n)+i] = P[$node 1 has a pointer at Level $L(n)+i$ or node 2 has a pointer at Level $L(n)+i$ or .. $]$ $\le$ $P[$node 1 has a pointer at Level $L(n)+i] + P[$node 2 has a pointer at Level $L(n)+i] +$ ..

$= n\cdot P[$arbitrary node has a pointer at level $L(n)+i] = n\cdot p^{L(n)+i}$

Hence, $C \le \sum_{i=1}^{ML-L(n)+i} n\cdot p^{L(n)+i-1} = \sum_{i=1}^{ML-L(n)+i} p^{i-1}$ (since $p^{L(n)} = \frac{1}{n}$)

$= \sum_{i=0}^{ML-L(n)-1} p^i \le \sum_{i=0}^{\infty} p^i = \frac{1}{1-p}$

Then, $A+B+C \le \frac{L(n)-1}{p} + \frac{1}{p} + \frac{1}{1-p} = \frac{L(n)}{p} + \frac{1}{1-p}$

$= \frac{\log_{1/p} n}{p} + \frac{1}{1-p} = O(\log n)$