Skip Lists

Probability of a node being at level $k > 1$ is

$$p^{k-1}(1-p)$$

> If p is the probility of having level $k+1$ wrt k for a skip list.

> Since all nodes at least have level 0, we don't calculate probability for the given level.

> So expected value of a nodes level is

$$1 + \frac{p}{1-p}$$

> Expected time complexity for search.

This can be calculated as expected length of search path. This starts at the top left node whose level (l) = list.maxlevel

It ends at a goal node at level $k \geq 0$ below l.

> we split the path starting from the goal node into two
  - The path to the top level
  - The path connecting the top left node and first top node we got in reverse.

Let $L(k)$ be expected length of first path. So for a node k levels below top we go back up one level with probability p or move left with probability $1-p$.

$$L(k) = 1 + pL(k-1) + (1-p)L(k)$$

$$pL(k) = 1 + pL(k-1)$$

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

Since $L(0) = 0$ (already at top left) we get

$$L(l) = \frac{l}{p}$$

We know the probability of a node being at level k so we expect top level to contain $p^{l-1}(1-p)n \leq p^l n$ nodes

Therefore, total expected time complexity of the search,

$$O\left(\frac{l}{p} + p^{l} n\right)$$

Let $p^{l} n = c$

$$p^{l} = c n^{-1}$$

$$l = \log_{p^{-1}}(c n^{-1})$$

$$l = \log_p c + \log_p n^{-1}$$

$$l = \log_p c + \log_{1/p} n \in O(\log n) + 1$$

The total expected complexity will be logarithmic if we set l to logarithm of n.
or total nodes level should be of the order of logarithm of

AVL Trees (Balanced search Trees)

> We calculate the balance factor for each node by subtracting height of right subtree from height of left subtree.

> |balance factor| ≤ 1

> Since height of an AVL tree is of the order

$$h = O(\log(N_n))$$

N_n - number of nodes

> So worst case time complexity for search is traversing $\log(n)$ levels so

search $\in O(\log n)$

In conclusion Skip lists provide almost the same search complexity as balanced AVL trees, but without the added complexity of rotations during insertion and deletion.