Expected cost of Skip Lists

Theorem 1. *Worst-case expected runtime for searching, inserting or deleting in skip lists is* $\Theta(\log n)$*.*

Proof. Let us consider the search path starting from the top left-most corner and ending at the largest key $<k$ on level 0. This path is made of "drop down" and "move right". Note that the complexity of our algorithms is Θ of the number of nodes in this path.

In fact we are going to consider the backwards path. This path starts from level 0 and rises to the top left-most corner by a series of left or up steps. The rule of this path is that it has to go up if it can (if you are not already at the top of the current tower) and otherwise it goes left.

Recall that skip lists are randomized data structures. We want to know the expected number of nodes in the backwards path considering all the random possibilities of skip lists for given keys.

First, assume that our skip list has height h. Let $T(h)$ be the worst-case expected number of nodes in the backward path. Here h also denotes the number of levels we must rise.

At any node in the backwards path, we have two options:

1. we can go up (and we do so). This happens with probability $1/2$ since it is related to a heads in a coin toss.

   Then we will have to rise one less level and
   $$\#\,\text{nodes} \leqslant 1 + T(h-1).$$

2. we can't go up and so we go left. This happens with probability $1/2$. In this case we still have to rise h levels and
   $$\#\,\text{nodes} \leqslant 1 + T(h).$$

So
$$T(h) \leqslant \frac{1}{2}\,(1 + T(h-1)) + \frac{1}{2}\,(1 + T(h))$$
which implies $T(h) \leqslant 2 + T(h-1)$ and so $T(h) \in \Theta(h)$.

Considering the fact that the expected height is $\Theta(\log n)$, we get that the expected number of nodes in our search path is $\Theta(\log n)$ and we deduce the $\Theta(\log n)$ worst-case expected complexity for operations on skip lists. $\square$