Analysis of Skip List

The length of search path in a Skip List is the sum of the lengths of search paths in each level.

\[ L = l_0 + l_1 + l_2 + \ldots + l_k \]

\[ E[L] = \sum_{i=1}^{k} E[\leq l_i] = \sum_{i=1}^{k} E[l_i] \]

\[ E[l_i] = 2 \quad (\text{Expected coin tosses to obtain the first head}) \]

This calculation is not dependent on \( x \), neither on the elements present in the Skip List.

\[ E[L] = 2k \quad (\text{the # of horizontal steps}) \]
At vertical steps = k

What is k? (depends on how many nodes survive at each level)

2 ways
1. We try to bound the expected value of k
   (the top level has \leq \alpha elements)

2. We choose k to be \, c \log n and then try to bound the expected # elements in level k, say?

Time to search \mid L_k \mid + \ldots

1. What is the prob that an element is promoted to \, c \log n levels, i.e. \, k \geq c \log n

   \[ \left( \frac{1}{2} \right)^{c \log n} = \frac{1}{nc} \]

Suppose we choose c = 2

Then the prob that any of the n elements survive c \log n levels \leq \frac{1}{nc} + \frac{1}{nc} \ldots \alpha times

Prob(\cup E_i) \leq \sum Prob(E_i) \leq \frac{1}{n} (Union bound)
If more than \( c \) elements survive 2 \( \log n \) levels, then we rebuild the data structure (Approach 2):

\[
E[K] = \sum_{i \geq 1} \left( \prod_{j=1}^{i-1} \Pr[X \geq j] \right) \quad \text{for a discrete r.v. } X
\]

So \( E[K] = \sum_{i \geq 1} \frac{1}{ni} \cdot \frac{1}{2^i} \cdot \frac{1}{n_i} \cdot \frac{1}{2^i} \cdot \frac{1}{n_i} = O(\log n) \)

Second approach:

\[
Y_i = \begin{cases} 
1 & \text{if } i \text{th element survives until } \log n \text{ levels} \\
0 & \text{otherwise}
\end{cases}
\]

\[
Y = Y_1 + Y_{2^1} \cdot Y_n
\]

\[
E[Y] = \sum E[Y_i] = \sum \Pr[\text{ith element survives}]
\]

\[
\leq \sum_{i=1}^{\log n} \frac{1}{nc} < 1
\]

Expected time for search = \( \frac{\text{expected no of elements in level } c \log n}{2 \cdot c \log n} \)
Concatenable queue (union and split) can be implemented in $O(\log n)$ time using a very simple procedure.