Skip Lists

Ben Napier

2020-2021

A *skip list* is a probabilistic data structure that allows $O(\log n)$ search complexity and insertion complexity within an ordered sequence of n elements.

Idea: for a sorted list, we create a duplicate list where a given element of the initial list is duplicated with probability 0.5. For every duplicated item, we store a pointer at the item in the second list back to the first list. We also add *sentinel nodes* for safety. Take the sorted list $(0, 1, 2, 3, 4, 5, 6)$ as follows.

$$\begin{tikzcd}
-\infty \arrow[r] & 0 \arrow[r] \arrow[d] & 1 \arrow[rr] \arrow[d] & & 3 \arrow[rrr] \arrow[d] & & & 6 \arrow[r] \arrow[d] & \infty \\
-\infty \arrow[r] & 0 \arrow[r] & 1 \arrow[r] & 2 \arrow[r] & 3 \arrow[r] & 4 \arrow[r] & 5 \arrow[r] & 6 \arrow[r] & \infty
\end{tikzcd}$$

Searching for 5 is shown below.

q

$$\begin{tikzcd}
-\infty \arrow[r] & 0 \arrow[r] \arrow[d] & 1 \arrow[rr] \arrow[d] & & 3 \arrow[rrr] \arrow[d] & & & 6 \arrow[r] \arrow[d] & \infty \\
-\infty \arrow[r] & 0 \arrow[r] & 1 \arrow[r] & 2 \arrow[r] & 3 \arrow[r] & 4 \arrow[r] & 5 \arrow[r] & 6 \arrow[r] & \infty
\end{tikzcd}$$

The probability a given duplicated node is followed by k non-duplicated nodes is $\frac{1}{2^k}$. Thus, the expected number of nodes examined when searching the initial list is $\sum_{k=1}^{\infty} \frac{1}{2^k} = 2$. So by adding this *shortcut list*, we have reduced the time complexity from n to $\frac{n}{2} + O(1)$.

An improvement: keep adding shortcut lists of the top shortcut list until we are out of elements.

$$\begin{tikzcd}
-\infty \arrow[rrrrrrrr] \arrow[d] & & & & & & & & \infty \arrow[d] \\
-\infty \arrow[rr] \arrow[d] & & 1 \arrow[rrrrrr] \arrow[d] & & & & & & \infty \arrow[d] \\
-\infty \arrow[rr] \arrow[d] & & 1 \arrow[rrrrr] \arrow[d] & & & & & 6 \arrow[r] \arrow[d] & \infty \arrow[d] \\
-\infty \arrow[r] \arrow[d] & 0 \arrow[r] \arrow[d] & 1 \arrow[rr] \arrow[d] & & 3 \arrow[rrr] \arrow[d] & & & 6 \arrow[r] \arrow[d] & \infty \arrow[d] \\
-\infty \arrow[r] & 0 \arrow[r] & 1 \arrow[r] & 2 \arrow[r] & 3 \arrow[r] & 4 \arrow[r] & 5 \arrow[r] & 6 \arrow[r] & \infty
\end{tikzcd}$$

The expected number of levels is $\log n$, which is easy to prove. At each level, we cut search time in half (excluding overhead). This gives us a search time of $O(\log n)$.