Skip Lists

Joshua Shim

October 2024

Insertion:

To insert k:

1. Search for right place to insert.
2. Toss a coin until you get a head, and count the number of tails t that you got.
3. Insert k and link it at levels 0-t from the bottom up.

Deletion: This is the same as a standard doubly linked list, except all layers must be taken care of.

Other Notes: Search:

1. The probability of getting i consecutive tails when flipping a coin is i times is $(\frac{1}{2})^i$.
2. If n elements belong to a set with a probability p each, then the expected size of that set is np.
3. Thus, an n element Skip List has on average $\frac{n}{2^i}$ elements with links on level i.
4. Since an element has links only on levels 0 - i with probability $\frac{1}{2^{i+1}}$, the total expected number of link levels per element is
$$\sum_{i=0}^{\infty} \frac{i+1}{2^{i+1}} = \sum_{i=1}^{\infty} \frac{i}{2^i} = 2$$
5. Let $\#(i)$ denote the number of elements on level i
6. Since the expected number of elements having a link at level i is $E[\#(i)] = \frac{n}{2^i}$, by the Markov inequality, the probability of having at least one element at level i satisfies $P(\#(i) \geq 1) \leq \frac{E[\#(i)]}{1} = \frac{n}{2^i}$
7. Thus, the probability to have an element on level $2 \log n$ is smaller than $\frac{n}{2^{2 \log n}} = \frac{n}{2^{\log n^2} = \frac{n}{n^2} = \frac{1}{n}}$.
8. More generally, the probability that level $k \log n$ is nonempty is smaller than $\frac{1}{n^{k-1}}$.

9. The expected value E OF k such that k is the least integer so that the number of levels is $\leq k \log n$ is

$$E \leq \sum_{k=1}^{\infty} \frac{k}{n^{k-1}} = (\frac{n}{n-1})^2$$

10. Thus, the expected number of levels is barely larger than $\log n$

11. If an element has a link at level i then with probability $\frac{1}{2}$, it has link at level $i+1$.

12. Thus, the expected number of elements between any two consecutive elements with a link on level $i+1$ which have links only up to level i is smaller than

$$\frac{0}{2} + \frac{1}{2^2} + \frac{2}{2^3} + \frac{3}{2^4} + \cdots = 1$$

So once on level i, on average we will have to inspect only two elements on that level before going to a lower level.

13. There will be fewer than $2\log(n)$ levels to go down, with visiting on average only two elements per each level.

14. Consequently, on average, the search will be in time $O(log(n))$.

15. For an element with links on levels $0-i$, we hve to store $2(i+1)$ pointers to other elements and the expected number of elements with highest link on level i is $O(\frac{n}{2^{i+1}})$. Thus, the total expected space for all pointers doe snot exceed

$$O(\sum_{i=0}^{\infty} 2(i+1)\frac{n}{2^{i+1}}) = O(2n\sum_{i=0}^{\infty} \frac{i+1}{2^{i+1}}) = O(4n) = O(n)$$

16. Unless we ensure that the worst case performance of search is $O(\log(n))$, Skip Lists are a better option than BSTs.