CSE 241 Class 14

Jeremy Buhler

October 19, 2015

Today: skip list analysis!

1 Things to Show

- Let H be the height of the tallest pillar in a skip list of size n.
- Want to show that H is "almost certainly" $O(\log n)$.
- Implies that likely cost of **delete** is $O(\log n)$.
- Moreover, will show that expected number of steps when traversing the list during a find is proportional to H.
- Will conclude that expected cost of **find** is "almost certainly" $O(\log n)$
- Similar argument holds for **insert**.

2 Bound on List Height

- Let h_j be height of jth node's pillar
- From before,
$$\Pr(h_j = t) = \left(\frac{1}{2}\right)^t$$
- What is $\Pr(h_j > t)$?

$$\begin{aligned}
\Pr(h_j > t) &= \sum_{i=t+1}^{\infty} \left(\frac{1}{2}\right)^i \\
&= \left(\frac{1}{2}\right)^{t+1} \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i \\
&= \left(\frac{1}{2}\right)^{t+1} \cdot \left(\frac{1}{1-1/2}\right) \\
&= \left(\frac{1}{2}\right)^t.
\end{aligned}$$

- What can we now say about $\Pr(H > t)$?

$$\begin{aligned}\Pr(H > t) &= \Pr(h_1 > t \text{ or } h_2 > t \text{ or } \ldots \text{ or } h_n > t) \\ &\leq \Pr(h_1 > t) + \Pr(h_2 > t) + \ldots + \Pr(h_n > t) \\ &= n \cdot \left(\frac{1}{2}\right)^t.\end{aligned}$$

Why did we use $\leq$ instead of $=$ above? This is example of the union bound, which is the simplest of the Bonferroni inequalities:

$$\begin{aligned}\Pr(h_1 > t \text{ or } h_2 > t) &= \Pr(h_1 > t) + \Pr(h_2 > t) - \Pr(h_1 > t \text{ and } h_2 > t) \\ &\leq \Pr(h_1 > t) + \Pr(h_2 > t)\end{aligned}$$

Graphically, can illustrate as

- OK, so in what sense is it "very likely" that H is $O(\log n)$?
- Note that

$$\begin{aligned}\Pr(h_j > c\log n) &= \left(\frac{1}{2}\right)^{c\log n} \\ &= \frac{1}{n^c}.\end{aligned}$$

- Moreover, same substitution gives

$$\Pr(H > c\log n) \leq \frac{1}{n^{c-1}}.$$

- In other words, the probability of H exceeding $c\log n$ goes down polynomially with n for any fixed c.
- Hence, large deviations of H above $c\log n$ for small c are very unlikely and only get less likely as n grows.

- In this sense, we say that H is "very likely" $O(\log n)$.
- **Defn**: let E_α be a probabilistic event parameterized by a number α. We say that E_α occurs *with high probability* if for $\alpha > 1$,

$$\Pr(E_\alpha) \geq 1 - \frac{k_\alpha}{n^\alpha}$$

where k_α is a constant that depends only on α, not on n.

- (In above example, E_α is "$H \leq (\alpha + 1)\log n$".)
- High-probability bounds are a stronger result than expected times, and they are preferred when studying randomized algorithms.
- (Skip list handout also proves directly that $E[H] = O(\log n)$; math is a bit icky, so we don't reproduce it here.)

3 Cost of Search

We will compute *expected* cost of a search, as full WHP result is again somewhat icky.

- Consider the trajectory taken by a search in the skip list.
- Can divide trajectory into horizontal and vertical "steps"

- Total search cost is proportional to number of steps.
- Sum of all vertical steps is $H - 1$ (start at height H, end at height 1).
- Can we get a bound on the number of horizontal steps?

To make things easier to see, we will run the search "backwards"!

- WLOG, we will assume that search always goes to the bottom of the skip list. (If we find the key early, drop down to the bottom level.)
- Backward traversal begins with a run of 0 or more horizontal moves from nodes of height 1.
- At some point, we encounter a node of height > 1, at which point we *must* take a vertical step up.

- (Why? If we pass a node of height > 1 at level 0 and only later jump up to a higher level, that would imply that the traversal algorithm went down when it could have moved forward to a node with a key less than the target.)

- Traversal then takes 0 or more horizontal moves from nodes of height 2, until it sees a node of height > 2.
- Repeat this pattern until we reach the head while traversing at height H.

Now, here's the key question.

- How many horizontal steps do we expect to take at a given level before taking a vertical step?
- Equivalently, how many nodes of height exactly t do we expect to encounter before the first node of height $> t$?
- Recall that heights are determined independently for every node by a geometric distribution with parameter 1/2.
- Given that a node has height at least t, the probability that it has height $> t$ is given by

$$\begin{aligned}
\Pr(h > t \mid h \geq t) &= \frac{\Pr(h > t \cap h \geq t)}{\Pr(h \geq t)} \\
&= \frac{\Pr(h > t)}{\Pr(h > t-1)} \\
&= \frac{(1/2)^t}{(1/2)^{t-1}} \\
&= 1/2.
\end{aligned}$$

- Conceptually, we can model the sequence of nodes of height $\geq t$ as a sequence of coin flips. Each tail is a node of height $= t$, while the first head is a node of height $> t$ that ends the sequence.
- Each tail causes one horizontal step.
- What is the number n_τ of tails before the first head?
- Chance that $n_\tau = j$ is $(1/2)^{j+1}$ for $j \geq 0$.

- Hence,

$$
\begin{aligned}
E[n_\tau] &= \sum_{j=0}^{\infty} j \left(\frac{1}{2}\right)^{j+1} \\
&= \frac{1}{2}\sum_{j=0}^{\infty} j \left(\frac{1}{2}\right)^{j}
\end{aligned}
$$

- Time to bust out a summation formula:

$$\sum_{j=0}^{\infty} jx^j = \frac{x}{(1-x)^2}$$

  for $0 < x < 1$.

- Plugging in $x = 1/2$, we get

$$
\begin{aligned}
E[n_\tau] &= \frac{1}{2}\frac{1/2}{(1-1/2)^2} \\
&= \frac{1}{2}\frac{1/2}{1/4} \\
&= 1.
\end{aligned}
$$

- Finally, the total number of steps N over the whole algorithm is given by $N = H - 1 + \sum_{t=1}^{H} n_\tau$ (since the analysis of horizontal steps holds at every level).

- Conclude that

$$
\begin{aligned}
E[N] &= H - 1 + \sum_{t=1}^{H} E[n_\tau] \\
&= H - 1 + \sum_{t=1}^{H} 1 \\
&= 2H - 1.
\end{aligned}
$$

- Since $H = O(\log n)$ WHP, we have that WHP, the expected number of steps taken by search is also $O(\log n)$.

One final thing: where did that summation formula come from?

- We know that

$$\sum_{j=0}^{\infty} x^j = \frac{1}{1-x}$$

  for $0 < x < 1$.

- Differentiate both sides w/r to x:

$$\sum_{j=0}^{\infty} jx^{j-1} = \frac{1}{(1-x)^2}$$

- Multiply both sides by x:

$$\sum_{j=0}^{\infty} jx^j = \frac{x}{(1-x)^2}$$