This handout gives the skip list methods that we discussed in class. A skip list is an ordered, doubly-linked list with some extra pointers that allow us to “jump” over multiple nodes at once when traversing the list in order. For a complete description of skip lists with some analysis, see the paper on the subject by their inventor, Bill Pugh, in *Communications of the ACM* 33(6):668–76, June 1990.

Each node in a skip list has a key and a *pillar* of some height \( t \) (which varies among nodes). A pillar is an array containing \( t \) pairs of *next* and *prev* pointers. Every pillar of height at least \( \ell \) is threaded into an ordered linked list by the pointers at level \( \ell \).

A skip list contains two “guard” pillars, the *head* and *tail*, which have key values \(-\infty\) and \(+\infty\) respectively. These pillars are at least as tall as any others in the list and so form the endpoints of the linked lists at every level.

Pillar heights are chosen randomly at the time of node insertion from a *geometric distribution* with parameter \( p = 1/2 \). This distribution is characterized by

\[
\Pr[\text{height} = t] = \left(\frac{1}{2}\right)^t
\]

for all \( t \geq 1 \). You can generate geometrically distributed random numbers by, e.g., repeatedly flipping an unbiased coin and counting the number of flips needed to obtain the first head. Although pillars can in theory be arbitrarily tall, the probability of creating a very tall pillar is extremely small; for example, there is less than one chance in a million that a given pillar will have height above 20.

In the following code, the procedure \textsc{RandomHeight()} returns a height according to the above geometric distribution. The notation \( x[\ell].\text{next} \) refers to the “next” pointer for the linked list at level \( \ell \) of \( x \)’s pillar.

## 1 Trivial Operations

The \textsc{min}, \textsc{max}, \textsc{succ}, and \textsc{pred} operations on skip lists are trivial (as they are for any ordered list). Successor and predecessor can be obtained for any node by following the *next* and *prev* links at the bottom (level 0) of its pillar. The min and max nodes are respectively the successor of the head and the predecessor of the tail. It is straightforward to recognize that, e.g., \textsc{succ} has reached the end of the list because it will encounter the tail node with value \(+\infty\).

Assuming we keep pointers to the list head and tail, all of these operations are \( \Theta(1) \).

Deleting a node \( x \) in a skip list requires that we unlink its pillar from the linked lists at every level. The pseudocode for this operation is:

\begin{verbatim}
REMOVE(x)
    for \ell in 0...x.height - 1 do
        splice x out of linked list at level \ell
\end{verbatim}

If the pillar to be deleted is of height \( t \), \textsc{Remove} requires time \( \Theta(t) \).
2 Nontrivial Operations

The two interesting operations on a skip list are FIND and INSERT. It will take some analysis to show that the expected cost of these operations is $O(\log n)$; for now, we just give the pseudocode.

The FIND operation traverses the list, as you might expect, but it tries to jump over as many nodes as possible with each step. Links at higher levels usually jump over more nodes than those at lower levels (because taller pillars occur less often), so we jump as far as we can at each level before descending to the level below it.

\[
\text{FIND}(k)
\]
\[
\ell \leftarrow \text{head.height} - 1 \\
x \leftarrow \text{head} \\
\text{while } \ell \geq 0 \text{ do} \\
\quad y \leftarrow x[\ell].\text{next} \\
\quad \text{if } y.\text{key} = k \\
\quad \quad \text{return } y \\
\quad \text{else if } y.\text{key} < k \\
\quad \quad x \leftarrow y \\
\quad \text{else} \\
\quad \quad \ell -- \\
\text{return null}
\]

Insertion is only slightly more interesting. We need to insert the newly allocated pillar at its correct place in the linked lists for every level. While we could traverse each linked list separately starting from the head node to find the right insertion point, we take advantage of the fact that if a new node $z$ should be inserted after a node $x$ at level $\ell$, then $z$ must also appear after (though perhaps not immediately after) $x$ at every level below $\ell$.

Note that we should start the insertion, like the FIND, from the top level of the skip list, not from level $t - 1$; otherwise, when $t = 1$ (half the time on average), we end up traversing $\Theta(n)$ nodes at level 0 to do the insertion.

\[
\text{INSERT}(z)
\]
\[
t \leftarrow \text{RandomHeight} \\
\text{allocate a pillar of height } t \text{ for } z \\
\ell \leftarrow \text{head.height} - 1 \\
x \leftarrow \text{head} \\
\text{while } \ell \geq 0 \text{ do} \\
\quad y \leftarrow x[\ell].\text{next} \\
\quad \text{if } y.\text{key} < z.\text{key} \\
\quad \quad x \leftarrow y \\
\quad \text{else} \\
\quad \quad \text{if } \ell < t \\
\quad \quad \quad \text{link } z \text{ into list at level } \ell \text{ between } x \text{ and } y \\
\quad \quad \ell --
\]

One slightly tricky detail of insertion is that we might allocate a new pillar larger than the head and tail pillars. When this happens, we can simply increase the height of the head and tail to accommodate the new node. As for resizable hash tables, it can be shown that if we double the head and tail height each time an
3 Performance Analysis

The performance analysis of skip lists depends heavily on the fact that their pillar heights are chosen from a geometric distribution. There are two things to show. First, the tallest pillar height, which upper-bounds the cost of deletion, isn’t too big; in particular, it’s very likely to be $O(\log n)$ for a skip list of $n$ elements. Second, a find operation in a skip list on average inspects a constant number of keys per level of the list; hence, the overall number of key inspections per search is also $O(\log n)$ on average.

3.1 Average Height of Tallest Pillar

The following proof of logarithmic expected height is much more precise (and more complex) than what we did in class. In class, we merely showed that, for some constant $c$, pillars of height $< c \log n$ were very likely, while those of height $> c \log n$ were very unlikely, suggesting that the “tipping point” between the two regimes is the typical behavior.

Let $H$ be the height of the tallest pillar in a skip list with $n$ elements (excluding the head and tail). First, it is not hard to see that

$$\Pr(H > t) \leq n \left(\frac{1}{2}\right)^t.$$ To see this, consider that the probability that $H > t$ is the probability that at least one pillar in the list has height $> t$. The latter is upper-bounded by $n$ times the probability that any single pillar has height $> t$.

We now wish to argue that $E[H] = O(\log n)$. By definition of expectation, we have

$$E[H] = \sum_{t=1}^{\infty} t \Pr(H = t).$$

We split this sum into several parts, each of which can be bounded independently:

$$E[H] = \sum_{t=1}^{2\lceil \log n \rceil} t \Pr(H = t) + \sum_{t=2\lceil \log n \rceil + 1}^{n} t \Pr(H = t) + \sum_{t=n+1}^{\infty} t \Pr(H = t).$$

(1)

For the first part, we may write

$$\sum_{t=1}^{2\lceil \log n \rceil} t \Pr(H = t) \leq 2 \lceil \log n \rceil \sum_{t=1}^{2\lceil \log n \rceil} \Pr(H = t) \leq 2 \lceil \log n \rceil \cdot 1.$$  

For the second part, we have that

$$\sum_{t=2\lceil \log n \rceil + 1}^{n} t \Pr(H = t) \leq n \sum_{t=2\lceil \log n \rceil + 1}^{n} \Pr(H = t) \leq n \Pr(H > 2 \lceil \log n \rceil) \leq n \left(\frac{n}{2^{2\lceil \log n \rceil}}\right) \leq n \left(\frac{n}{n^2}\right) = 1.$$
Here, we’ve simply plugged in our original bound on $\Pr(H > t)$.

Finally, for the third part, we use the following sum formula for $x < 1$:

$$\sum_{t=0}^{n} tx^t = \frac{x}{x-1} \left[ (n+1)x^n - \frac{x^{n+1} - 1}{x-1} \right]$$  \hspace{1cm} (2)

(You can derive this formula yourself in the same way as the formula for $\sum_{t=0}^{\infty} tx^t$ – see Appendix A of your text.) We have that

$$\sum_{t=n+1}^{\infty} t \Pr(H = t) \leq \sum_{t=n+1}^{\infty} t \Pr(H > t - 1) \leq 2n \sum_{t=n+1}^{\infty} t \left( \frac{1}{2} \right)^t = 2n \left[ 2 - \sum_{t=0}^{n} t \left( \frac{1}{2} \right)^t \right].$$

The third step uses our original upper bound for $\Pr(H > t)$, while the fourth uses the fact that

$$\sum_{t=0}^{\infty} tx^t = \frac{x}{(1-x)^2}.$$  \hspace{1cm} (Again, see Appendix A of your text.) Plugging $x = \frac{1}{2}$ into Equation (2), we find that

$$\sum_{t=0}^{n} t \left( \frac{1}{2} \right)^t = 2 - n \left( \frac{1}{2} \right)^n.$$

Conclude that

$$\sum_{t=n+1}^{\infty} t \Pr(H = t) \leq 2n \left[ 2 - \left( 2 - n \left( \frac{1}{2} \right)^n \right) \right] = 2n^2 \left( \frac{1}{2} \right)^n = o(1).$$

Combining our bounds for all three terms of the sum in (1), we conclude that

$$E[H] \leq 2 \lceil \log n \rceil + 1 + o(1),$$

implying that $E[H] = O(\log n)$.

### 3.2 Constant Expected Search Time Per Level

Consider a find operation in a skip list. The find function can be broken down into a series of traversals at different levels $\ell$ of the list, starting with the first key inspected at level $\ell$ and ending when a key inspection causes us to go down instead of forward. How many keys are inspected on average at level $\ell$?

To analyze this situation, we start by observing that every key inspected at level $\ell$, except possibly the last, belongs to a pillar of height exactly $\ell + 1$ (that is, a pillar whose maximum level is $\ell$). Suppose not, i.e. suppose that some key $k$ that is not the last has an associated pillar $p$ of hight $\ell + 1$. If $k'$ is the argument to the find function, we know that $k < k'$, since otherwise the level’s traversal would terminate at or before
Hence, the search will pass pillar $p$ at level $\ell$. But this means that the search would have passed pillar $p$ at a higher level (say, $\ell + 1$) as well, since $p$ is in higher-level chains and we never move backwards during a traversal. Hence, we would not actually inspect $p$’s key at level $\ell$. Conclude that our supposition is false, and $p$ must have height exactly $\ell + 1$.

We can now bound the expected length of the traversal at level $\ell$ by asking: how many pillars of height exactly $\ell + 1$ do you expect to traverse before encountering a pillar of height $> \ell + 1$? Each pillar at level $\ell$ is generated independently at random from a geometric distribution with parameter $1/2$, and all have height at least $t = \ell + 1$. For each such pillar, the chance that its height $h$ exceeds $t$ is given by

$$\Pr(h > t \mid h \geq t) = 1 - \Pr(h = t \mid h \geq t) = 1 - 1/2 = 1/2.$$  

To see the $1/2$, consider that the decision on whether to stop the pillar at height $t$ or continue to grow it corresponds to a single, unbiased “coin flip.”

Now the expected traversal length is one plus the expected number $E[n_t]$ of pillars of height exactly $t$ prior to the first pillar of height $> t$. We have probability $1/2$ that $n_t = 0$, i.e. that we immediately exceed height $t$ on the first new key. Similarly, $n_t = 1$ with probability $1/4$, 2 with probability $1/8$, and so forth. Conclude that

$$E[n_t] = \sum_{t=0}^{\infty} t \left(\frac{1}{2}\right)^{t+1}$$

$$= \frac{1}{2} \sum_{t=0}^{\infty} t \left(\frac{1}{2}\right)^{t}$$

$$= \frac{1}{2} \cdot \frac{1/2}{(1 - 1/2)^2}$$

$$= \frac{1}{2} \cdot \frac{1/2}{(1 - 1/2)^2}$$

$$= 1.$$

Hence, if we include the key that terminates the traversal at level $\ell$, the average length of this traversal is $1 + 1 = 2$.

Since we expect the number of levels traversed to be $O(\log n)$, the total expected number of key inspections is also $O(\log n)$.