How do you find min? \( \text{head}.\text{next}[0] \)

max? \( \text{tail}.\text{prev}[0] \)

How do you move to next/prev element from a given tower? \( \text{ptr}.\text{next}[0] \) \( \text{ptr}.\text{prev}[0] \)
How do your search for an element?

level

head

peak

tail

would fall here

"∞" a a b c r s t t "∞"
L₂: r
L₁: b r t
L₀: a a b c r s t t

Tower<E> findFirstOccurrence(E target) {
    Tower<E> left = head;
    Tower<E> right = tail;
    int level = height - 1;
    while (level ≥ 0) {
        Tower<E> next = left.next(level);
        if (right == next)
            level--;
        else {
            if (comp.compare(target, next.element) > 0)
                left = next;
            else {
                right = next;
                level--;
            }
        }
    }
    return right;
}
How can you find predecessor/successor of an element (e.g. cl)?

Use level 0 to navigate in sorted order

"-∞" a a b c e r s t t "∞"
Insertion

level

head

tail

let $p$ be prod of a tail (continue).
Ex. $p=\frac{1}{4}$

Find position to insert using search

Then randomly pick height of new tower as # biased coin flips until head obtained
Once the position is found and height is selected, just splice new tower into list for each level (already know where from search).

Also update height (and occasionally must resize head/tail).
Deletion

Search to find + then splice out of each level it is in (just like in a linked list).

Also update height (+ occasionally must resize head/tail).