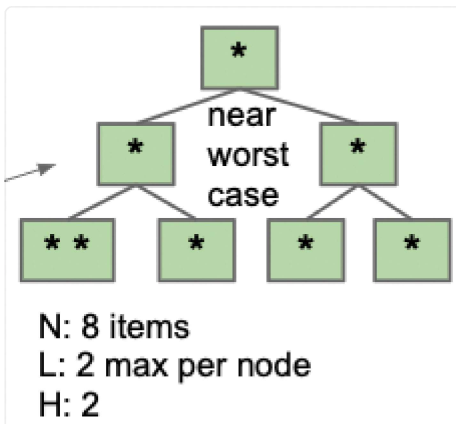
17.5 B-Tree Performance

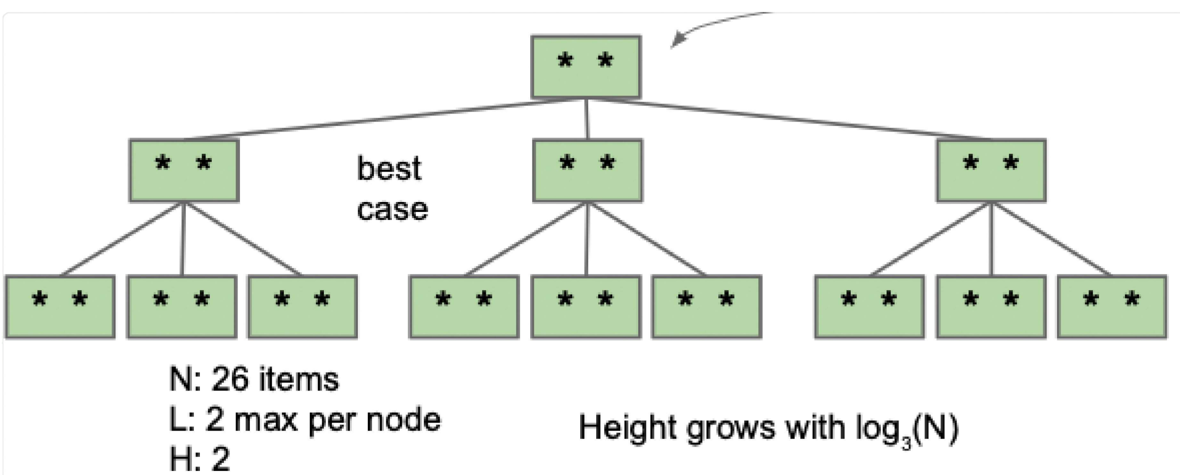
B-Tree Runtime Analysis

To consider the runtime of B-Trees, let L be the maximum items per node. Based on our invariants, the maximum height must be somewhere between $\log_{L+1} N$ (best case, when all nodes have L items) and $\log_2 N$ (worst case, when each node has 1 item).

The overall height, then, is always on the order of $\Theta(\log N)$



Worst-case B-Tree height



Best-case B-Tree height

Runtime for `contains`

In the worst case, we have to examine up to L items per node. We know that height is logarithmic, so the runtime of `contains` is bounded by $O(L \log N)$. Since L is a constant, we can drop the multiplicative factor, resulting in a runtime of $O(\log N)$.

Runtime for `add`

A similar analysis can be done for `add`, except we have to consider the case in which we must split a leaf node. Since the height of the tree is $O(\log N)$, at worst, we do $\log N$ split operations (cascading from the leaf to the root). This simply adds an additive factor of $\log N$ to our runtime, which still results in an overall runtime of $O(\log N)$.

Previous
17.4 B-Tree Invariants

Next
17.6 Summary