B+Tree Complexity

B+Tree Complexity

The B+Tree is an n-ary tree which is height balanced, hence search is going to be $O(\log n)$

In practice, search will take $log_m n$ reads of pages where the fanout ratio of the tree is m (The fanout ratio is the number of children in each node). With a 4KByte block, a 4 byte integer key, a 4 byte disk address size, with every block being half full, and even assuming a little space is in each block is taken with extra administrative information, that gives a fanout ratio of at least 250.

- Tree of height 1: 250 records
- Tree of height 2: 62,500 records
- Tree of height 3: 15,625,000 records
- Tree of height 4: 3,906,250,000 records

B+Tree Complexity

This is actually an underestimate because the nodes will, on average have more entries in them and the block sizes used will usually be larger than 4Kbytes

Thus can find a record from a collection of approx 4 Trillion in 4 disk reads.

In practice, we would normally cache all except the bottom two levels in memory, so really only takes 2 disk reads.

Note that the cost of the disk read is so much larger than the cost of the processing of in-memory aspects of the data structure that we can ignore in-memory processing costs.

Insertion has the same order of complexity as search.