B-Tree Delete Complexity Proof

Shyamal Suhana Chandra

1 Theorem: B-Tree Delete Complexity

Statement: Deleting from a B-Tree with n keys takes $O(\log n)$ time.

2 Proof

1. Find node to delete: $O(\log n)$
2. If internal node, find predecessor/successor: $O(\log n)$
3. Delete from node: $O(1)$
4. Handle underflow:
   - Borrow from sibling: $O(1)$
   - Merge with sibling: $O(t) = O(1)$
   - Propagate upward: at most $O(\log n)$ levels
5. Total time: $O(\log n) + O(\log n) = O(\log n)$

Conclusion: B-Tree delete has $O(\log n)$ time complexity.