B-Tree Space Complexity Proof

Shyamal Suhana Chandra

1 Theorem: B-Tree Space Complexity

Statement: A B-Tree with n keys uses $O(n)$ space.

2 Proof

- Each key is stored exactly once: n keys
- Each node has at most $(2t-1)$ keys and $2t$ children pointers
- Number of nodes: at most $\frac{n}{t-1} = O(n)$ when t is constant
- Total space:

$$\begin{aligned} \text{Space} &= n \text{ keys} + O(n) \text{ pointers} && (1)\\ &= O(n) + O(n) && (2)\\ &= O(n) && (3) \end{aligned}$$

Conclusion: B-Tree space complexity is $O(n)$.

3 Detailed Analysis

For a B-Tree of order t with n keys:

- Minimum nodes: $\frac{n}{2t-1}$ (when all nodes are full)
- Maximum nodes: $\frac{n}{t-1}$ (when nodes have minimum keys)
- Each node stores: at most $(2t-1)$ keys and $2t$ pointers
- Total storage: $O(n)$ keys + $O(n)$ pointers = $O(n)$

Since t is typically a small constant (e.g., $t = 3$ to $t = 100$), the space overhead is linear in the number of keys.