B^+ 树查找和插入的复杂度分析

B^+ 树插入的第一步是找到插入的叶节点。这个步骤的复杂度与树高和节点中路径查找相关。

M 阶的 B^+ 树，节点元素个数最少是 $\lceil \frac{M}{2} \rceil$。因此树高的最大值为：

$$Depth(M,N) = O(\lceil \log_{\lceil \frac{M}{2} \rceil} N \rceil) = O(\frac{\log N}{\log M})$$

每个节点元素是 M，有序存放，二分查找路径的复杂度为 $O(\log M)$。因此

$$T_{Find}(M,N) = O(\log M \times \frac{\log N}{\log M}) = O(\log N)$$

插入新元素有可能引发节点分裂。最坏情况是由根节点向叶节点一路分裂上去。每个节点分裂的复杂度是 $O(M)$（M 个元素拷贝到新节点），因此插入分裂的最坏情况是：

$$T(M,N) = M \times \frac{\log N}{\log M} = O((M/\log M)\log N)$$