This problem considers the String B-tree, as described in [KärkkäinenRao03]. Theorem 7.3 states that search among $N$ strings for strings with prefix $P$ can be done in

$$O(\text{search}(N) + \text{scan}(|P| + Z))$$ I/Os,

where $Z$ is the number of results. However, it is not clear from the description how one avoids having to compare $P$ to each of the $Z$ results, which would take $O(\text{scan}(|P|)Z)$ I/Os.

1. In the description of the search algorithm on page 160 one case is left out. What is the missing detail? (Hint: Is it always the case that only one child contains strings with prefix $P$?)

2. Given a Patricia trie and a leaf in the trie having prefix $P$, how does one identify all leaves with prefix $P$, without accessing $P$ or any string in the Patricia trie? The length of $P$ is known.

3. Show that search in a String B-tree can be done in $O(\text{search}(N) + \text{scan}(|P| + Z))$ I/Os. Note that results may span several nodes in the String B-tree.