Week 14 Summary: Prefix Tries, KMP, and Rabin-Karp String Matching

Z5007 Programming and Data Structures · MTech DSAI · Odd Semester 2026-27 · Dr. Innocent Nyalala

1. Prefix Tries for Dictionary and Autocomplete

A Trie branches on characters rather than whole keys, supporting insertion, exact search, and prefix lookup in $O(L)$ time for word length L, independent of dictionary size.

2. Knuth-Morris-Pratt (KMP) Linear-Time Search

Naïve matching restarts from scratch on mismatches, costing $O(nm)$ on repetitive inputs such as `aaaa...b`. KMP precomputes the Longest Proper Prefix-Suffix table $\text{lps}[0..m-1]$ in $\Theta(m)$ time so mismatches jump the pattern cursor to $\text{lps}[j-1]$ without moving the text cursor backward, achieving $\Theta(n+m)$ worst-case time.

3. Rabin-Karp Rolling Hash

Represent pattern P and each m-character window as base-d integers modulo prime q. Shifting the window one step right updates the hash in $O(1)$ arithmetic operations, checking character equality only on hash collisions.

Worked Check

For pattern `ababd`, $\text{lps} = [0, 0, 1, 2, 0]$. Searching `pemba_clove_pemba_cassava_pemba` for `pemba` locates exact indices $[0, 12, 26]$ in one linear pass.