Rolling Hash is invented by Robin Karp.

Rolling hash means next hashCode(bcd) is hashCode(abc) – a + d

No recalculation of hashCode bcd because it is derived from hashCode of abc.

On internet there is single pattern sample code. But Rolling hash single pattern search is inferior to KMP and Boyer-moore b/c it does not use jump. It moves character a time like brute force. Rolling Hash’s advantage is multiple pattern match.

Wikipedia proposes a multiple pattern search pseudo code but it is only good for patterns with same length which is not useful.

We figured out way to implement multiple patterns with different length as demo will show. It is much useful.

Our approach is to use the shortest length of all patterns to build hashTable for initial lookup. Should there is indication of potential match, we then compare the substring with the original pattern by using java’s “equalsIgnoreCase()”.

Store hashCode of shortened patterns and then original patterns in hashTable

HashTable<String, String> patternHashTable = new HashTable<String, String>();

patternHashTable.**put**(“234”, “swimming”);

Initial detection of potential match with the shortened pattern:

If (patternHashTable.**get**(hashCodeOfSubstring) ! = null )

Do character by character with java.equalsIgnoreCase()

BigO should be better than brute force

Brute force for multiple pattern would be O(n \* M \* K)

BigO should be:

O(n + m + k ) - n (length of text ), m (length of pattern ), k ( number of output matches )

Similar to **Aho–Corasick algorithm.**

The result is stored in a hashTable:

HashTable<Integer, String> results = new HashTable<Integer, String>()

Location of first character, original pattern string

The result is used to turn font to red in the result display panel.

GUI has 3 sections:

1. Pattern and search button ( JTextArea and JButton)

The button’s event handler calls run()

Run() creates RollingHash object and pass in pattern set

The pattern set filters out special characters and common short patterns/pronouns

It is best to search words that are longer than 3 characters as short pattern like ‘is’ going to cause a lot of match ( false alarm )

1. The “text to be searched” panel ( JTextArea )

We can copy and paste any text into it

1. “Result display area” panel

It uses JTextPane which allows us to change font and color by indicating which index location, how long is substring and what color.

The search result hashTable provides location index and pattern(length)