Working mod $q = 11$, how many spurious hits does the Rabin-Karp matcher encounter in $S = 3141592653589793$ when looking for $P = 26$

3

Knuth-Morris-Pratt

- Best known for linear time for exact matching
- Compares from L-R but shifts more than one position
- Preprocessing appraoch of pattern to avoid trivial comparisons
- Conceived by Donald Knuth and Vaughan Pratt
- Guaranteed worst-case efficiency $\Theta((n-m+1)m)$
- Preprocessing time is $O(m)$
- Searching time is $O(n)$

Two main aspects:

Pre-processing Involves parsing through the pattern alone ($O(m)$) time and space- an array of prefix-suffix match is created

Searching Involves parsing though the string using the pre-processed array and the patern array ($O(n)$) time

$T = AAABAAAB, P = AAAA$

- Compute in advance how far to jump in P (Pre-processing) when a match fails
- Retain information from prior attempts
- Never decrement i, ever.

  The length of the longest proper prefix in the (sub)pattern that matches a proper suffix in the same (sub)pattern

$$LPS[0,?,?,?]$$
$$LPS[0,1,?,?]P = \mathbf{AA}AA$$
$$LPS[0,1,2,?]P = \mathbf{AAA}A$$
$$LPS[0,1,2,3]P = \mathbf{AAAA}$$

$P = AAABAAAA$

$$\begin{gathered}
LPS[0,?,?,?,?,?,?,?] \\
LPS[0,1,?,?,?,?,?,?]P = \mathbf{A}AABAAAA \\
LPS[0,1,2,?,?,?,?,?]P = \mathbf{AA}ABAAAA \\
LPS[0,1,2,3,?,?,?,?]P = \mathbf{AAA}BAAAA \\
LPS[0,1,2,3,3,?,?,?]P = \mathbf{AAA}BAAAA \\
LPS[0,1,2,3,3,3,?,?]P = \mathbf{AAA}BAAAA \\
LPS[0,1,2,3,3,3,3,?]P = \mathbf{AAA}BAAAA \\
LPS[0,1,2,3,3,3,3,3]P = \mathbf{AAA}BAAAA
\end{gathered}$$

How do we get $LPS[7]$ from $LPS[6]$?

$$LPS[6] = 2 : P = \mathbf{AA}AB\mathbf{AA}AA$$

Check if $P[7] = P[3]$: Yes $LPS[7] = LPS[6] + 1$

$P = ABACABAB$ $LPS = [0,0,1,0,1,2,3,2]$ $LPS[7] = 3 : P = \mathbf{ABA}C\mathbf{ABA}B$
Check if $P[8](B) = P[4](C)$ No! (Not able to create longer P/S) We can make use of the info $[0,0,\mathbf{1},0,1,2,3,2]$ Meanining $P[1] = P[7](P[1] = P[3] = P[5] = P[7])$