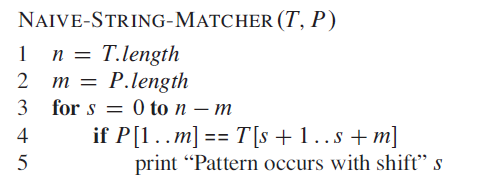
**String manipulation/matching algorithms: Rabin Karp algorithm**

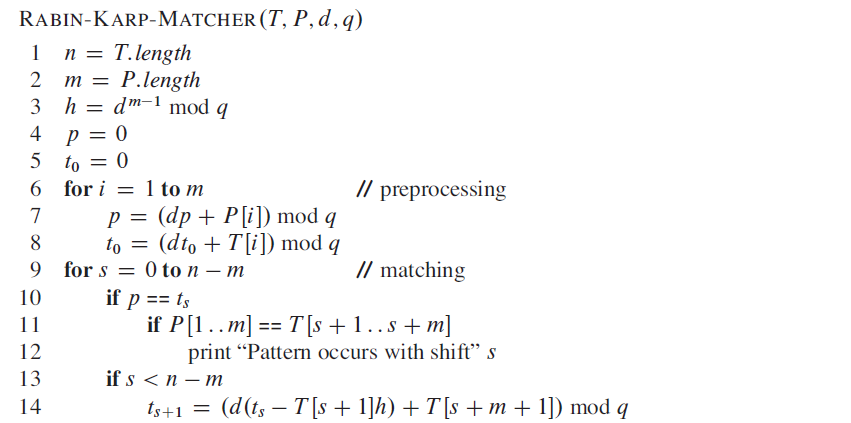
**Naïve brute-force algorithm:**

Naïve pattern searching is the simplest method among other pattern searching algorithms. It checks for all character of the main string to the pattern. This algorithm is helpful for smaller texts. It does not need any pre-processing phases. We can find substring by checking once for the string. It also does not occupy extra space to perform the operation.

****

**The Rabin-Karp-Algorithm**

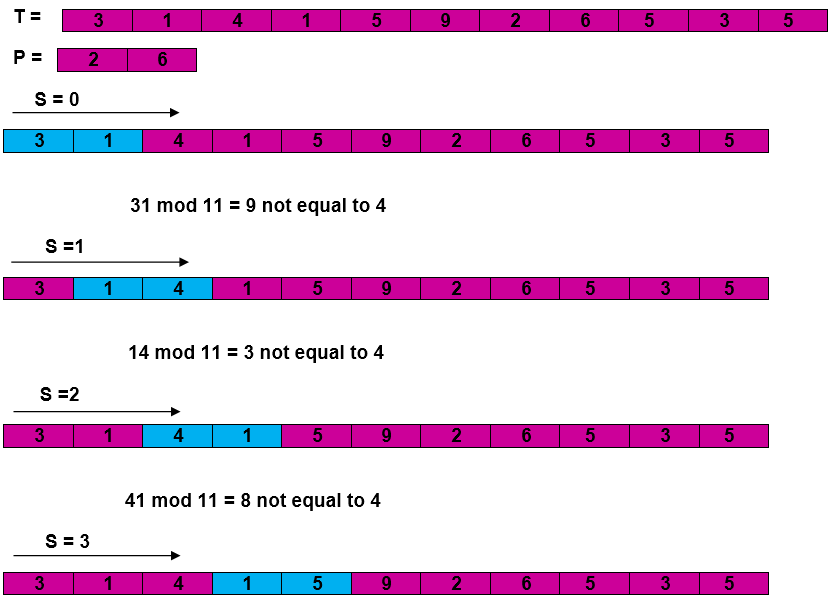
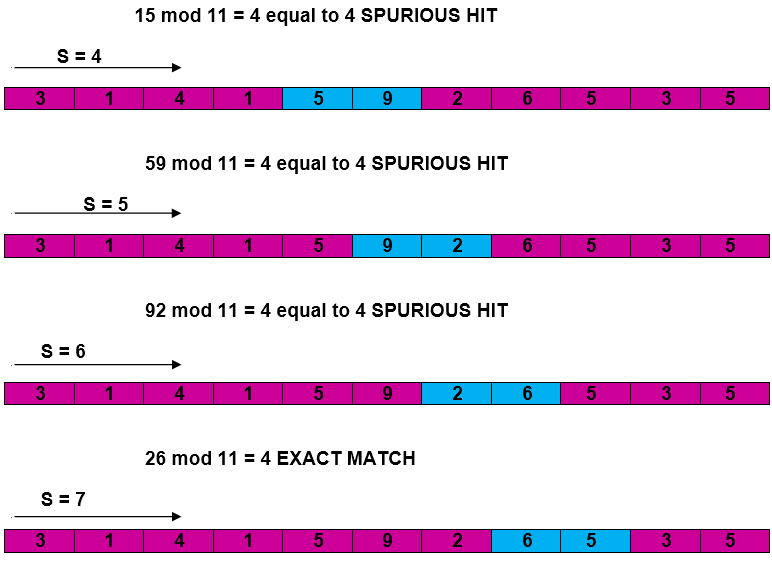
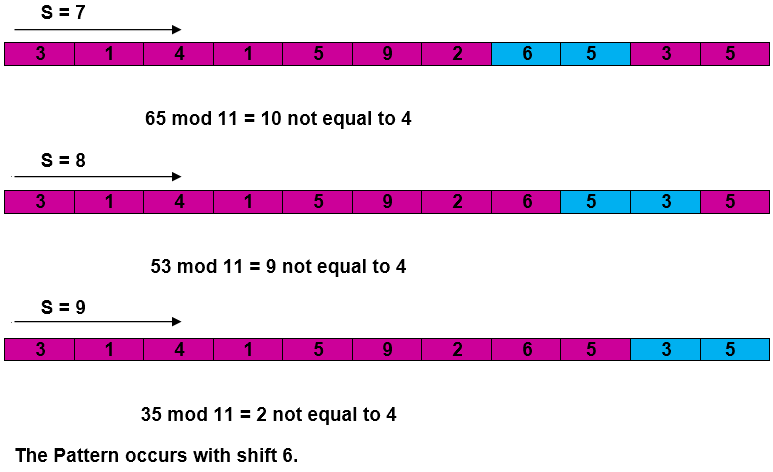
The Rabin-Karp string matching algorithm calculates a hash value for the pattern, as well as for each M-character subsequences of text to be compared. If the hash values are unequal, the algorithm will determine the hash value for next M-character sequence. If the hash values are equal, the algorithm will analyze the pattern and the M-character sequence. In this way, there is only one comparison per text subsequence, and character matching is only required when the hash values match.

****

**Example:** For string matching, working module q = 11, how many spurious hits does the Rabin-Karp matcher encounters in Text T = 31415926535.......

1. T = 31415926535.......
2. P = 26
3. Here T.Length =11 so Q = 11
4. And P mod Q = 26 mod 11 = 4
5. Now find the exact match of P mod Q...

**Solution:**

## **Complexity:**

The running time of **RABIN-KARP-MATCHER** in the worst-case scenario **O ((n-m+1) m** but it has a good average case running time. If the expected number of strong shifts is small **O (1)** and prime q is chosen to be quite large, then the Rabin-Karp algorithm can be expected to run in time **O (n+m)** plus the time to require to process spurious hits.

**RELEVANT READING MATERIAL AND REFERENCES:**

**Source Notes:**

1. <https://www.tutorialspoint.com/Naive-Pattern-Searching#:~:text=Na%C3%AFve%20pattern%20searching%20is%20the,is%20helpful%20for%20smaller%20texts>.
2. <https://www.javatpoint.com/daa-rabin-karp-algorithm#:~:text=The%20Rabin%2DKarp%2DAlgorithm,for%20next%20M%2Dcharacter%20sequence>.

**Lecture Video:**

1. https://youtu.be/crY4nrN6A9w

**Online Notes:**

1. <http://vssut.ac.in/lecture_notes/lecture1428551222.pdf>

**Text Book Reading:**

1. Cormen, Leiserson, Rivest, Stein, “*Introduction to Algorithms*”, Prentice Hall of India, 3rd edition 2012. problem, Graph coloring.

**In addition: PPT can be also be given.**