Assignment No. 10

Aim: Write C++ Program to demonstrate the implementation of Rabin-Karp Algorithm with discussion of time complexity.

Theory:

- **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.

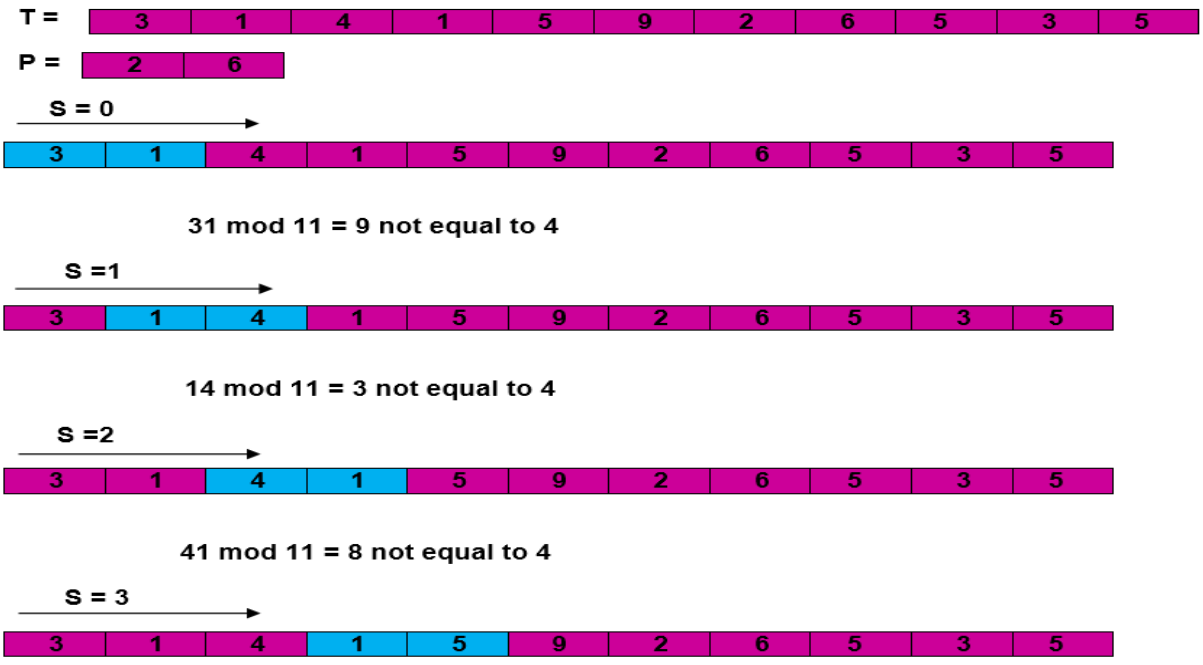
- **Example:**

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

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

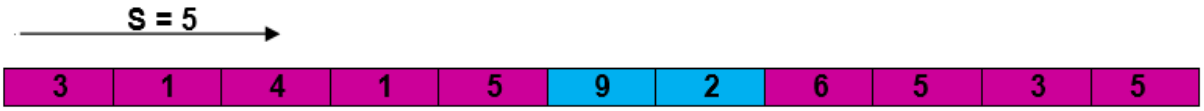
Solution:

The solution of the Rabin karp Algorithm is given as belows and its functioning is explained here as well , look for the correct placing and the memory processing in the algorithm.

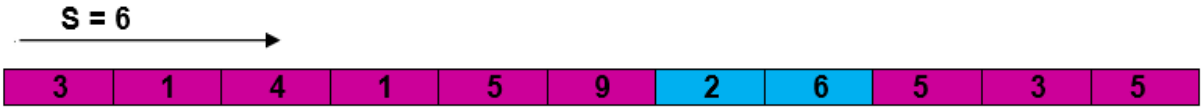


15 mod 11 = 4 equal to 4 SPURIOUS HIT

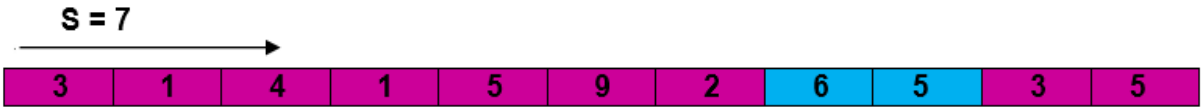
59 mod 11 = 4 equal to 4 SPURIOUS HIT

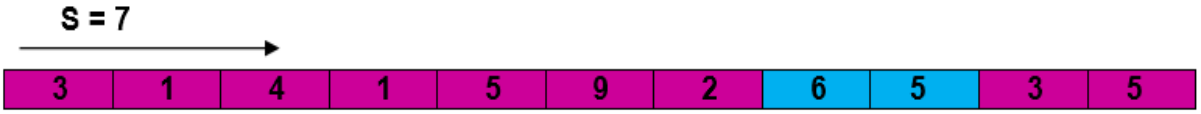


92 mod 11 = 4 equal to 4 SPURIOUS HIT

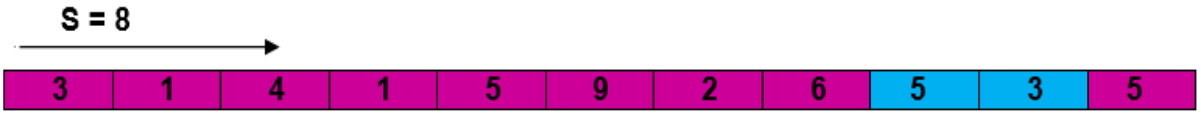


26 mod 11 = 4 EXACT MATCH

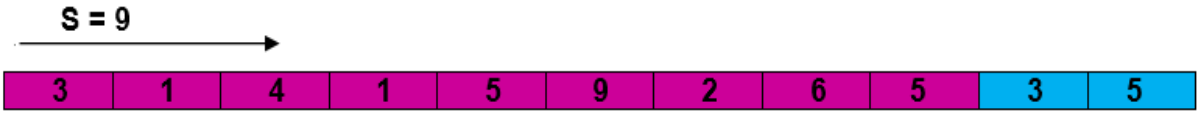




65 mod 11 = 10 not equal to 4



53 mod 11 = 9 not equal to 4



35 mod 11 = 2 not equal to 4

The Pattern occurs with shift 6.

- **Complexity of the Algorithm:**

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.

- **Further usage:**

Rabin-Karp algorithm is an algorithm used for searching/matching patterns in the text using a hash function. Unlike Naive string matching algorithm, it does not travel through every character in the initial phase rather it filters the characters that do not match and then performs the comparison.

Program:

Output:

Conclusion: