Rabin-Karp String Searching

- A string search algorithm which compares a string's hash values, rather than the strings themselves.

How Rabin-Karp works

- Let characters in both arrays T and P be digits in radix-Σ notation. ($\Sigma = (0,1,...,9)$)
- Let p be the value of the characters in P
- Choose a prime number q such that fits within a computer word to speed computations.

How Rabin-Karp works (continued)

- Compute (p mod q)
  - The value of p mod q is what we will be using to find the pattern P in T.
- Compute (T[s+0, .., s+m-1] mod q) for s = 0 .. n-m
- Test against P only those sequences in T having the same (mod q) value

Rabin-Karp Algorithm

```
Algorithm  RabinKarp( char T[], char P[],  int n, int m)
  {  // The inputs are pattern P, input string T
     h^P = hash(P)              // m characters of given Pattern P
     h^T = hash(T[0..m-1]   //first m characters from text T
     for S= 0 to n-m do
        {
         if(h^P  == h^T )    // Two hash values are compared
             {     If (P[0..m-1]==T[S+0 .. S+m-1])
                   {     Print " Pattern Found with shift  S"
                      return
                   }
                else { // try in the next level }
             }
          if (S<n-m)
                h^T = hash(T[S+1..S+m]
       }
          Print " Pattern not found "
  }
```

A Rabin-Karp example

- Given T = 31415926535 and P = 26
- We choose q = 11
- P mod q = 26 mod 11 = 4

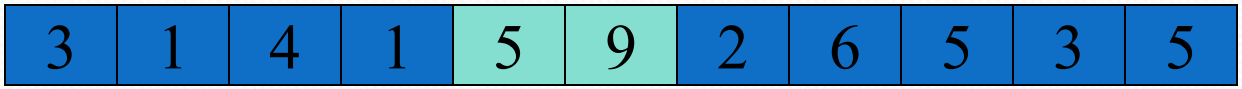
31 mod 11 = 9 not equal to 4

14 mod 11 = 3 not equal to 4

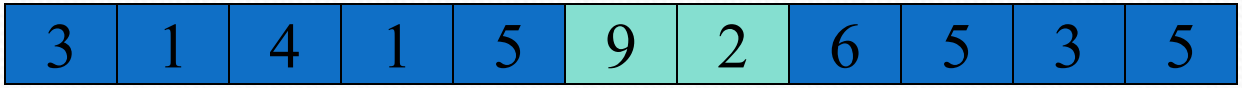
41 mod 11 = 8 not equal to 4

Rabin-Karp example continued

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 equal to 4 -> an exact match!!