Data Structures and Algorithms

Prof. Ganesh Ramakrishnan,
Prof. Ajit Diwan,
Prof. D.B. Phatak

Department of Computer Science and Engineering
IIT Bombay

Session: Rabin-Karp Algorithm

Introduction: Rabin-Karp Algorithm[1]

- Assume that $\sum = \{0, 1, 2, ..., 9\}$, where each character is a decimal digit in radix-d i.e $d = \left|\sum\right|$
- A string of k consecutive characters represents a length-k decimal number
- To find all occurrences of the pattern $P[1..m]$ in the text $T[1..n]$
  - Compute hash ($number \mod q$, where q is a prime) of the number P of size m and compare it with m consecutive digits of T
  - If the two hash numbers match, check for each digits of T and P of size m
  - Report the occurrence of pattern P in text T
  - Can be generalized to any set of characters

[1] Chapter 32, CLRS, Third Edition

Illustration

$$P = \underbrace{8 \quad 4 \quad 7 \quad 2 \quad 6}_{\text{mod } 17 = 15}$$

$$Text = 3 \quad 8 \quad 4 \quad 7 \quad 2 \quad 6 \quad 3 \quad 9 \quad 5 \quad 1 \quad 7$$

Illustration

$$\underbrace{3\ 8\ 4\ 7\ 2}_{\text{mod } 17 = 1}\ 6\ 3\ 9\ 5\ 1\ 7$$

Invalid Match

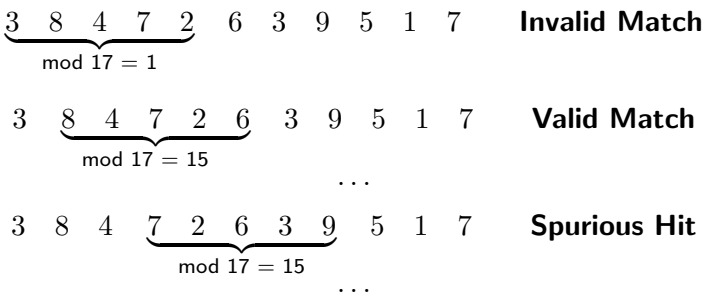
Illustration

$\underbrace{3\ 8\ 4\ 7\ 2}_{\text{mod } 17 = 1}\ 6\ 3\ 9\ 5\ 1\ 7$ **Invalid Match**

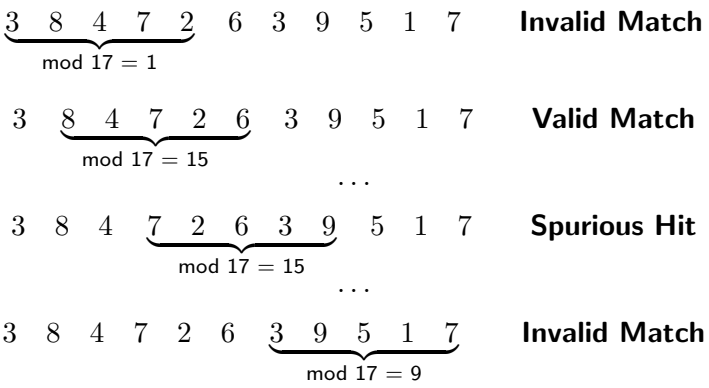
$3\ \underbrace{8\ 4\ 7\ 2\ 6}_{\text{mod } 17 = 15}\ 3\ 9\ 5\ 1\ 7$ **Valid Match**

$\ldots$

Illustration



Illustration



Computing rolling hash value

$$\underbrace{3\ \ 8\ \ 4\ \ 7\ \ 2}_{\text{mod } 17 = 1}\ \ 6 \qquad\qquad 3\ \ \underbrace{8\ \ 4\ \ 7\ \ 2\ \ 6}_{\text{mod } 17 = 15}$$

- To compute hash value of 84726, for radix d = 10, and prime q = 17
  - Old high order digit = 3
  - New low-order digit = 6

$$\begin{aligned} 84726 &= (38472 - 3 * 10000) * 10 + 6(\ mod\ 17) \\ &= 15\ mod\ 17 \end{aligned}$$

Rabin-Karp Algorithm

Algorithm Rabin-KarpAlgorithm(T, P, d, q)
Input Text T of size n, Pattern P of size m, the radix d, and prime q
Define s as the shift index to T
$h = d^{m-1} \mod q$
$p = 0, t_0 = 0$
for $i \in (0...m-1)$ **do**
    $p = (d * p + P[i]) \mod q$
    $t_0 = (d * t_0 + T[i]) \mod q$
end for
...
... Continued on next slide

Figure: Rabin-Karp Algorithm

Rabin-Karp Algorithm

```
...
...
for s ∈ (0...n − m) do
    if p = t_s then
        j = 0
        while j < m & T[s + j] = P[j] do
            j = j + 1
        end while
        if j = m then
            print 'Valid at shift s'
        end if
    end if
    if s < n − m then
        t_{s+1} = (d(t_s − T[s]h) + T[s + m + 1])mod q
    end if
end for
```

Figure: Rabin-Karp Algorithm

Analysis of Rabin-Karp Algorithm

Algorithm Rabin-KarpAlgorithm(T, P, d, q)
Input Text T of size n, Pattern P of size m, the radix d, and prime q
Define s as the shift index to T
$h = d^{m-1} \mod q$
$p = 0, t_0 = 0$
for $i \in (0...m-1)$ **do**
    $p = (d * p + P[i]) \mod q$
    $t_0 = (d * t_0 + T[i]) \mod q$
end for $\implies$ $c_1 \times m$ times (preporcessing, computing hash)
...
... Continued on next slide

Figure: Analysis of Rabin-Karp Algorithm

Analysis of Rabin-Karp Algorithm

```
...
...
for s ∈ (0...n − m) do
    if p = t_s then
        j = 0
        while j < m  &  T[s + j] = P[j] do
            j = j + 1
        end while   ⟹   c_2   × m times
        if j = m then
            print 'Valid at shift s'
        end if
    end if
    if s < n − m then
        t_{s+1} = (d(t_s − T[s]h) + T[s + m + 1])mod q
    end if
end for   ⟹   c_3   × (n − m + 1) times (matching)
```

Figure: Analysis of Rabin-Karp Algorithm

$$T(n) = c_1 m + c_2 c_3 (n - m + 1)m = \Theta((n - m + 1)m)$$

Thank you