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]

$t_s = T[s+1, \dots s+m]$

$= T[s+m] + 10(T(s+m-1) + 10(T(s+m-2) \dots 10T[s+1]))$

a^n , a^m $\Rightarrow \Theta((\underbrace{n-m+1}_{T})\underbrace{m}_{P})$

$T \nearrow$, $P \nearrow$

d-values

- Assume that $\sum = \{0, 1, 2, ..., 9\}$, where each character is a decimal digit in radix-d i.e $d = |\sum|$

radix d (d=10)

- 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

$T[s, \dots s+m-1]$

$P[1, \dots m]$

$p = P[m] + 10(P(m-1) + 10(P(m-2) \dots 10P(1)))$

$p \bmod q = t_s \bmod q$

[1] Chapter 32, CLRS, Third Edition

Illustration

$\Sigma = \{0, 1 \ldots 9\}$ $q = 17$

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

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

Illustration

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

Invalid Match

$p \bmod q \neq t_s \bmod q \Rightarrow p \neq t_s$

Illustration

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

Invalid Match

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

Valid Match

...

$p \bmod q = t_{s+1} \bmod q \longrightarrow$ verify that $p = t_{s+1}$

Illustration

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

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

$\dots$

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

$\dots$

$\underbrace{p \bmod q = t_{s+3} \bmod q}_{\text{false alarm / spurious hits } = O\left(\frac{n}{q}\right)}$ but $p \neq t_{s+3}$

Illustration

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

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

$\ldots$

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

$\ldots$

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

Computing rolling hash value

Horner's rule: $\underbrace{t_{s+1}}_{\text{mod}} = 10\underbrace{\left(t_s - 10^{m-1} T[s+1]\right) + T[s+m+1]}_{\text{mod}}$

$h = 10^{m-1} \bmod q$

left shift

removal of 3

adding digit at rightmost end

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

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

Modular arithmatic: $(a \bmod q)(b \bmod q) \bmod q = (ab) \bmod q$

$$84726 = (38472 - 3 * 10000) * 10 + 6(\ mod\ 17)$$
$$= 15\ mod\ 17$$

Rabin-Karp Algorithm

$|\Sigma| = d$

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

$p = P[0, .. m-1]$
$t_0 = T[0, \ldots m-1]$

Figure: Rabin-Karp Algorithm

Rabin-Karp Algorithm

...
...
for $s \in (0...n-m)$ **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

ensure $P[0,..,m-1] = T[s,...s+m-1]$

j j

$T[s,..s+m-1] \bmod q$

$d^{m-1} \bmod q$

dynamically update t_{s+1}

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 $\Longrightarrow$ c_1 $\times$ m times (preporcessing, computing hash)
...
... Continued on next slide

Figure: Analysis of Rabin-Karp Algorithm

Analysis of Rabin-Karp Algorithm

worst case $T = a^n$ $P = a^m \rightarrow O((n-m+1)m)$

```
...
...
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 ⟹ c2 × 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 ⟹ c3 × (n − m + 1) times (matching)
```

scan on T

c spurious hit

$O((n-m+1) + cm)$

Average: $O(n/q)$ spurious hit

$O(n)$

Figure: Analysis of Rabin-Karp Algorithm

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

Thank you

Thank you

Thank you

Thank you