Algorithms Design and Analysis [ETCS-301]

Dr. A K Yadav
Amity School of Engineering and Technology
(affiliated to GGSIPU, Delhi)
akyadav1@amity.edu
akyadav@akyadav.in
www.akyadav.in
+91 9911375598

November 1, 2019

Rabin-Karp algorithm I

- The algorithm depends on the fact that two numbers will be equal if and only if the modulus of these two numbers to a third number is equal.
- If two numbers are same then the modulus of these numbers to a third number will be same or if the modulus of the two numbers are not equal then these two numbers can not be same.
- If the modulus of the two numbers are equal then numbers may or may not be same.
- For string matching we will find out the modulus of text T and pattern P treating as numerical value to a third number and if the remainder is not same then no need to check the two strings.

Rabin-Karp algorithm II

- But if the remainder of the text T and pattern P to a third number is same then text T may or may be equal to pattern P. So we will check $P \sqsupset T_{s+m}$.
- Let p is numerical value of the pattern $P[1..m]$ in suppose radix-d(10).
- Let t_s denote the decimal value of the length-m substring of text $T[s+1..s+m]$ for $s = 0$ to $n-m$
- $t_s = p$ if and only if $T[s+1..s+m] = P[1..m]$
- So s is a valid shift if and only if $t_s = p$
- We can calculate t_0 and p in time $\Theta(m)$ using Horners rule:

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

$P[m]$ is the least order digit and $P[1]$ is the highest order digit.

$$t_0 = T[m]+10(T[m-1]+10(T[m-2]+\cdots+10(T[2]+10T[1]))\dots)$$

Rabin-Karp algorithm III

- The remaining $n - m$ values $t_1, t_2, \ldots, t_{n-m}$ can be computed in $\Theta(n - m)$ time each requires constant time as per the below formula:

$$t_{s+1} = 10(t_s - 10^{m-1} T[s + 1]) + T[s + m + 1]$$

Algorithm I

RABIN-KARP-MATCHER(T,P,d,q)

1. $n = lenght(T)$
2. $m = lenght(P)$
3. $h = d^{m-1} mod\ q$
4. $p = 0$
5. $t_0 = 0$
6. for $i = 1$ to m
7. $\quad p = (dp + P[i]) mod\ q$
8. $\quad t_0 = (dt_0 + T[i]) mod\ q$
9. for $s = 0$ to $n - m$
10. $\quad$ if $p == t_s$
11. $\quad\quad$ if $P[1..m] == T[s + 1..s + m]$
12. $\quad\quad\quad$ print Pattern occurs with shift s

Algorithm II

13. if $s < n - m$

14. $t_{s+1} = (d(t_s - T[s+1]h) + T[s+m+1])mod\ q$

Complexity of the algorithm is $\Theta(m)$ for preprocessing of the string P and T_m to find out p and t_0 respectively. For all $t_0, t_1, \ldots, t_{n-m}$, we check $P[1..m] = T[s+1..s+m]$ if modulus of p and t_i for $0 \leq i \leq n - m$ to a third number is same. So total time taken is $\Theta((n - m + 1)m)$.

Thank you

Please send your feedback or any queries to akyadav1@amity.edu, akyadav@akyadav.in or contact me on +91 9911375598