

Karp Rabin Algorithm

Rajesh Addanki

December 14, 2011

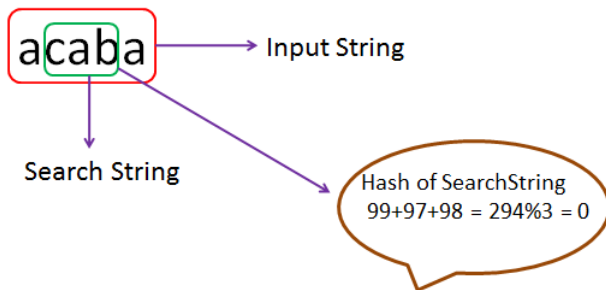
Introduction

- Karp Rabin Algorithm is a string searching algorithm, It is one of the effective string matching algorithms.
- It uses technique called hash .
- Hash technique used to find any set of patterns in a string.

Karp-Rabin Algorithm is created by Miachel O.Rabin and Richard M.Karp in the year 1987.

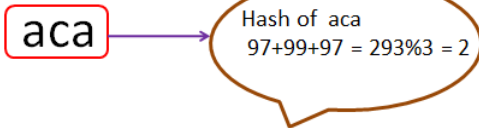
Example

Let us consider an input string



Step 1: Initially we will consider the first three characters that is `aca`.

Step 1:



Now we will consider `cab`.

Step 2:

`cab`



Hash of cab
 $99+97+98 = 294\%3 = 0$

Now we will consider aba.

Step 3:

aba



Hash of cab
 $97+98+97 = 292\%3 = 1$

- In the above scenarios, step2 result matches with search string hash function, check whether the string matches with search string . Search String matches
- Result is 1,3
1 is the iteration number and 3 is the length of search string.

cab = cab

Karp-Rabin Algorithm

```
 $p \leftarrow \text{inputstring}$   
 $q \leftarrow \text{searchstring}$   
 $n \leftarrow \text{length}(p)$   
 $m \leftarrow \text{length}(q)$   
 $h\text{SrchStr} \leftarrow \text{hash}(q)$   
for  $i = 0 \rightarrow n - m$  do  
     $h\text{Str} \leftarrow \text{hash}(p[i \rightarrow m])$   
    if  $p[i \rightarrow m] = q$  then  
        return "FOUND"  
    else  
        return "NOTFOUND"  
    end if  
end for
```

Calculating the hash

$Str \leftarrow \text{inputstring}$

$sum \leftarrow 0$

$y \leftarrow \text{length}(Str)$

$x[] \leftarrow Str$

for $i = 0$ **to** y **do**

$sum = sum + str[i]$

end for

"return $sum \% 3$ **"**

- Preprocessing $O(m)$ time complexity with constant space.
The preprocessing phase of the Karp-Rabin algorithm consists in computing $\text{hash}(q)$.
- Searching in $O(nm)$ time. During searching phase, it is enough to compare $\text{hash}(q)$ with SrchStr . If an equality is found, it is still necessary to check character by character.
- Expected run time is $O(n + m)$ that is by number of text character comparisons.

Comparisons

Comparisons with other string matching algorithms are

- Knutt-Morris-pratt
- Boyer Moore

Knutt-Morris-pratt





- It performs comparisons from left to right.
- Preprocessing $O(m)$ time complexity as the length of the longest border of the input string followed by a character.
- Searching phase in $O(n + m)$. The Knuth-Morris-Pratt algorithm performs at most $2n-1$ text character comparisons during the searching phase and the delay (maximal number of comparisons for a single text character). It is independent from the alphabet size.

- Preprocessing $O(m + n)$ time complexity with constant space. The searching phase time complexity is quadratic but at most $3n$ text character comparisons are performed when searching for a non periodic pattern. On large alphabets (relatively to the length of the pattern) the algorithm is extremely fast.
- Searching in $O(nm)$ time. It is the worst case.
- $3n$ text character comparisons in the worst case when searching for a non-periodic pattern.
- The best case is $O(n/m)$, which is the absolute minimum for any string-matching algorithm in the model where the pattern only is preprocessed.

Features

- It is independent of length of pattern
- It results accurately.
- Karp-Rabin algorithm is used for multiple pattern search
- Application of Karp-Rabin is detecting Plagiarism

Bibliography

-  Graham A Stephen, "*String Searching Algorithms*", World Scientific Calculate Publishers,(2000).
-  Dan Gusfield, "*Algorithms on Strings,trees, and Sequences*", Press Syndicate of the University on Cambridge(1997).
-  Jun-Ichi-Aoe, "*Computer Algorithms- String Pattern Matching Strategies*", IEEE Computer Society Press(1994).
-  Richard M.Karp, Miachel O.Rabin, "*Efficient Randomized Pattern Matching Algorithms*", Volume 31(1987),pages:249-260.