European Journal of Molecular & Clinical Medicine

ISSN 2515-8260 Volume 06, Issue 01, 2019

ASCII KM Discrete Matcher: Enhancement of ASCII KS Discrete Matcher using Pattern with Variable Length

Kuljinder Singh Bumrah[1], Himani Sivaraman[2], Dr. Sandeep Budhani[3]

Department of Computer Science and Engineering ,

Graphic Era Hill University

[1]*kuljinder.cse07@gmail.com*

[2]*himanisivaraman@gmail.com*

[3]*sandeepbudhani13@gmail.com*

Abstract: **It has been observe in past years that the academics have instigated string matching algorithms in several turfs to extract and find the undisclosed key RSA algorithm or any other encryption method for discovering the intruder or malicious pattern in imposition detection system , the DNA matching, carbon chain matching . This article has a signified aim to analyse ,suggest and attain an algorithm which is classified for discrete pattern matching in different mode and moreover more efficiently and effectively. It has been designed for any pattern or type of string whether a character string , decimal value , wild character or any special symbol using your ASCII value.**

Keywords: **ASCII, matching-algorithm, private key, encryption, pattern matching, cryptography**

INTRODUCTION:

A pattern correspondence or a string matching problem is a defined classified operation of finding one or the entire existence of that pattern of characters 'p' of length 'm ' in large text 't' of having a length 'n' in a discrete format . The observation has been extensively premediated[1]. The string -matching or the pattern matching are mostly used in information retrieval, bibliographic search, molecular biology, cryptography and interrogation responding application[2][4]. String matching is a very significant issue in the province of text processing and its algorithmic manifestation is measured as the rudimentary component used in implementation of practical software pattern under most operating system. So moreover, they have emphasized programming methods that serve as a paradigm in other dimensions of Computer Science[3]. As we all know the ASCII characters are the standard representation of 7 bit for each character. There are several larger character sets that use 8 bits, which gives them 128 additional characters. The extra characters are used to represent non-English characters, graphics symbols, and mathematical symbols.

THE DISCRETE PATTERN MATCHING AND THE STRING PATTERN MATCHING

Theoretically , if we define pattern and code they are different in there meaning. Patterns are a sequence distinctly organized manner[7]; the occurrence of pattern can be visualize anywhere. Code is carrier of information, it should convey a meaning . The pattern matching algorithms works on this basic scenario[1]. The pattern Algorithms search for a generalize string in a given doc /word file/browser, There are many types of algorithm available today like Naïve, KMP and many more.

String matching or would say searching algorithms usually follow a technique of finding place of one or several strings (patterns) [4][7] are found within a searched text(string). Both the pattern and the searched text are the amalgamation of the element of the alphabet i.e finite set. If we talk about the DNA matching and carbon chain matching it usually contains the fixed pair double helical structure.

String search algorithm	Time complexity for	
	preprocessing	matching
Naïve	0 (none)	$\Theta(n \cdot m)$
Rabin-Karp	$\Theta(m)$	avg $\Theta(n+m)$ worst $\Theta(n \cdot m)$
Finite state automaton	$\Theta(m\|\Sigma\|)$	$\Theta(n)$
Knuth-Morris-Pratt	$\Theta(m)$	$\Theta(n)$
Boyer-Moore	$\Theta(m+\|\Sigma\|)$	$\Omega(n/m)$, $O(n)$
Bit based (approximate)	$\Theta(m+\|\Sigma\|)$	$\Theta(n)$

Table 1. shows the time complexity for many string searching algorithm in accordance to pre-processing and matching credibility.

NEW PERCEPTION

The Discrete method matching is commenced in this resesrch. Pattern is fragmented in segment of one, or two, or three or so on character and ASCII value is calculated and then compared[1]. The ASCII value of each character of core text is compared with ASCII value of pattern character by character, at the end the temporary array contains the position of keys found in main string[1]. If the length of temporary array is exactly equal to the length of pattern then the pattern is found else pattern not found in the main string in discrete as well as in contiguous manner.

ASCII KM Discrete Matcher Algorithm : We have generated an effective Algorithm for the new perception created by changing the ASCII value of each character of the foremost text is compared with the ASCII value of the pattern of the main string[2]. The algorithm has been designed as below.

```
i←0
j←0
m←length[P]
while S!=NULL
        a←ASCII_VALUE( P[ j ] )
        b←ASCII_VALUE( S[ i ] )
        if a=b
            arr[ j ]←i
            j←j+1
        i←i+1
if m=j
```

```
        print "Pattern Found"
else
        print "Pattern Not Found"
ASCII_VALUE(char ch)
    return(int(ch))
```

COMPARISION:

The International Journal "ASCII KS: Discrete matcher" has some disadvantage[1] :

- It takes the fixed length of the pattern to be searched.
- It searches with the set of two sub characters of the pattern.
- It may create the problem if, the length of pattern to be searched is in odd number character in it.
- The complexity of the algorithm is very high.

This paper has improved all these problems:

- This algorithm has variable length of the pattern to be searched.
- It searches with variable sets of character of pattern. It searches even if the one character is found in the data Set (String or text).
- It does not create problem, even if whatever the length of the patter to be searched.
- The complexity of the algorithm is (O (n)).

CONCLUSIONS

A foremost algorithm for discrete pattern method is commence rated. In the mentioned algorithm again ASCII value is used to make it possibly work the algorithm for any type of string. Pattern is broken in any number of character segments. This algorithm is the improvement of the ASCII KS Discrete Matcher Algorithm which uses the two character segments, but this improved algorithm works for any character segment. This algorithm has very less complexity (O(n)) in comparison with ASCII KS Algorithm.

REFERENCES

[1] ASCII KS Discrete Matcher, A International journal of Computer Application.

[2] Research article an efficient ASCII-Based algorithm for single pattern Matching.

[3] Knuth, D, Morris, J.H, Pratt, V, Fast Pattern matching in strings, SIAM Journal on Computing, Vol. 6, No. 2, pp.323-350, 1977.

[4] D.E.Knuth, J.H.M.Jr., and V.R.Pratt, Fast pattern matching in strings,.SIAM J.Comput., vol. 6,no. 2, pp. 323.350, 1977.

[5] R.S.Boyer and J.S.Moore, Afast string matching algorithm, Commun. ACM, Vol.20, no. 2,October 1977.

[6] Z.Galil and J.Seiferas, Time space optimal string matching, in STOC, 1981.

[7] Karp, R.M, Rabin, M.O, Efficient randomized pattern matching algorithm, IBM J.Res. Dev., Vol. 31,No. 2, pp.249-260, 1987.

[8] D.M.Sunday, Avery fast sub string search algorithm, vol. 33, no. 8, pp. 132.142, August 1990.

[9] M.Crochemore and D.Perrin, Tow way string matching, J.ACM, vol. 38, no. 3, pp. 650-674, 1991.

[10] N.T, Deterministic memory efficient string matching algorithms for intrusion detection, INFOCOM, 2004.

[11] Greg Plaxton Theory in Programming Practice, Fall 2005 Department of Computer Science University of Texas at Austria.

[12] Introduction to Algorithm by Thomas H. Cormen.

[13] S Lipschutz and M.Lipson, Discrete Mathematics.

[14] A.Aho and M.Corasick. Efficient string matching:an aid to bibliographic search.

[15] A.Amir, A.Butman, and M.Lewenstein. Real scaled matching. In Proc. SODA 2000, page to appear, San Francisco, January 2000.

[16] A.Amir and G.Calinescu Alphabet independent and dictionary scaled matching. In Proc. CPM'96, number1075 in LNCS,pages 320-334, 1996.

[17] A.Amir and M.Farach Efficient 2-dimentional approximate matching of non – rectangular figure. InProc. SODA'91, page 212-223,1991.