

# ASCII KS Discrete Matcher: A Different and a New Approach to String Matching using Pattern in Discrete Manner

Kuljinder Singh  
Bumrah  
Graphic Era Hill  
University,  
Bhimtal

Saurabh Rawat  
Graphic Era University,  
Dehradun

Anushree Sah  
The University of  
Greenwich, U.K

Sumit Pundir  
The Graphic Era  
University,  
Dehradun

## ABSTRACT

Many researchers have implemented string matching algorithms in various fields for instance finding the secret key RSA or other encryption methods as well as finding intruder's pattern in Intrusion Detection System, DNA matching, Carbon Chain matching. This paper aims to analyze and obtain an algorithm for discrete pattern matching. It also works for any type of string whether it be characters, decimal values or special symbols.

## 1. INTRODUCTION

The string matching problem, also called pattern matching [2], is defined as the operation of finding one or all of the occurrences of a pattern of characters P of length m in a larger text T of length n in discrete form. The problem has been extensively studied. String matching algorithms are mostly used in information retrieval, bibliographic search, molecular biology, cryptography and question answering applications. String matching is a very important issue in the domain of text processing [1] and its algorithms are considered as the basic components used in implementations of practical software under most operating systems. Moreover, they emphasize programming methods that serve as paradigms in other fields of computer science.

## 2. A New Concept – ASCII KS Discrete Matcher:

All In this algorithm pattern matching is done in discrete manner [12]. Pattern is broken in a segment of two characters and ascii value is calculated of each character and hash value of each segment is calculated by multiplying the second character of the segment by 255 because the maximum ascii value can exceed to 255. The hash value of each segment is stored in an array. Then the ascii value of each character of the main text is calculated and hash value of pair of characters is calculated by shifting character –by-character and the hash value is stored in another array. The hash value of the two array is compared if all the value of the each segment of the pattern is matched with the each the main text, then the pattern is matched otherwise not. The algorithm has been designed (figure 1) below.

### ASCII KS Discrete Matcher

$n \leftarrow \text{length}[T]$

$m \leftarrow \text{length}[P]$

$j \leftarrow 0$

$k \leftarrow 0$

for  $i \leftarrow 1$  to  $m$

do  $a \leftarrow \text{ASCII\_VALUE}(P[i])$

$hp \leftarrow hp + a * \text{pow}(255, j)$

$j \leftarrow j + 1$

if  $j = 2$

$pt\_hash[k] \leftarrow hp$

$hp \leftarrow 0$

$k \leftarrow k + 1$

$j \leftarrow 0$

$l \leftarrow 0$

for  $s \leftarrow 1$  to  $n$

do  $ht \leftarrow 0$

$a \leftarrow \text{ASCII\_VALUE}(t[s])$

$b \leftarrow \text{ASCII\_VALUE}(t[s+1])$

$ht \leftarrow ht + a + b * 255$

$txt\_hash[l] \leftarrow ht$

$l \leftarrow l + 1$

$z \leftarrow 0$

for  $i \leftarrow 1$  to  $k$

```

for j←1 to l
    do if pt_hash[i]=txt_hash[k]
        do      f←1
            pos←j+1
            z←z+1
        else
            f←0
        if f=1
            do print “ Segment matches at
position”,i+1,pos
        else
            print “Segment does not
matches”
        if z=k
            do print “ Pattern Matches”
        else
            print “ Pattern Doesn’t Matches”
ASCII_VALUE(char c)
a←(int)c
return a

```

**Fig 1: Algorithm for Discrete Pattern Matching**

### 3. Conclusions

A new algorithm for discrete pattern matching is also designed. In this algorithm again ascii value is used so as to work the algorithm for any type of string. Pattern is broken in a segment of two characters and ascii value is calculated of each character and hash value of each segment is calculated by multiplying the second character of the segment by 255 because the maximum ascii value can exceed to 255. The hash value of each segment is stored in an array. Then the ascii value of each character of the main text is calculated and hash value of pair of characters is calculated by shifting character – by-character and the hash value is stored in another array. The hash value of the two array is compared if all the value of the each segment of the pattern is matched with the each of the main text, then the pattern is matched otherwise not.

### 4. REFERENCES

- [1] Research article An Efficient ASCII-Based Algorithm for Single Pattern Matching (Rami H.Mansi and Jehad Q. Alnihoud)
- [2] Knuth, D., Morris, J. H., Pratt, V., Fast pattern matching in strings, SIAM Journal on Computing, Vol. 6, No. 2, pp.323–350, doi:10.1137/0206024, 1977.
- [3] 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, June 1977.
- [4] R. S. Boyer and J. S. Moore, .A fast string matching algorithm,.Commun. ACM, vol. 20, no. 10, pp. 762.772, October 1977.
- [5] Z. Galil and J. Seiferas, .Time-space optimal string matching (preliminary report),. in STOC, 1981.
- [6] Karp, R. M., Rabin, M. O., Efficient randomized pattern-matching algorithms, IBM J. Res. Dev., Vol. 31, No. 2, pp. 249-260, 1987
- [7] D. M. Sunday, .A very fast substring search algorithm,. Commun. ACM,vol. 33, no. 8, pp. 132.142, August 1990.
- [8] M. Crochemore and D. Perrin, .Two-way string matching,. J. ACM, vol. 38, no. 3, pp. 650.674, 1991.
- [9] N. T. et.al., .Deterministic memory-ef\_cient string matching algorithms for intrusion detection,. INFOCOM, 2004.
- [10] Greg Plaxton Theory in Programming Practice, Fall 2005 Department of Computer Science University of Texas at Austin
- [11] Introduction to Algorithms by Thomas H. Cormen
- [12] S Lipschutz and M. Lipson, Discrete Mathematics
- [13] A. Aho and M. Corasick. Efficient string matching: an aid to bibliographic search. CACM, 18(6):333–340, June 1975.
- [14] A. Amir, A. Butman, and M. Lewenstein. Real scaled matching. In Proc. SODA 2000, page To appear.,San Francisco, January 2000.
- [15] A. Amir and G. Calinescu. Alphabet independent and dictionary scaled matching. In Proc. CPM’96,number 1075 in LNCS, pages 320–334, 1996.
- [16] A. Amir and M. Farach. Efficient 2-dimensional approximate matching of non-rectangular figures. InProc. SODA’91, pages 212–223, 1991.
- [17] A. Amir and G. Landau. Fast parallel and serial multidimensional approximate array matching. TheoreticalComputer Science, 81:97–115, 1991.
- [18] M. Atallah. A linear time algorithm for the Hausdorff distance between convex polygons. InformationProcessing Letters, 17:207–209, 1983.
- [19] R. Baeza-Yates. Similarity in two-dimensional strings. In Proc. COCOON’98, number 1449 in LNCS,pages 319–328, Taipei, Taiwan, August 1998.
- [20] R. Baeza-Yates and G. Navarro. Multiple approximate string matching. In Proc. WADS’97, LNCS1272, pages 174–184, 1997.