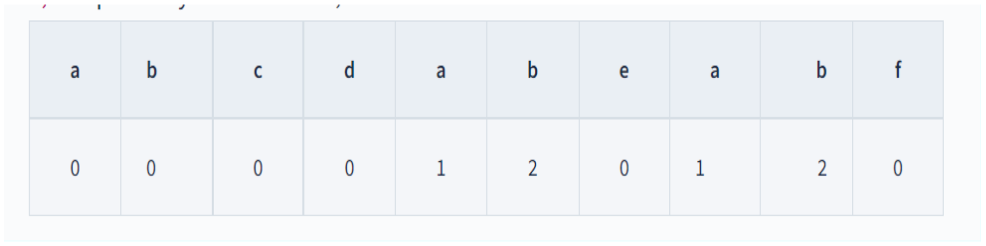
Assignment 1 – DSA

20001312

P.A.T.N.Perera

I used KMP algorithm for this problem. This algorithm compares character by character from left to right. When a mismatch happens, it uses the “Prefix table” to skip some characters previously comparison while matching. KMP algorithm is worst case efficient. Worst case time is o(n) in this. (In boyer moore – o(n) , Rabin carp – o(mn), String native- o(m(n-m+1)).) The preprocessing time is always o(n) and searching time is always o(m). The running time complexity of this algorithm is o (m+n).

Example for prefix table



a	b	c	d	a	b	e	a	b	f
0	0	0	0	1	2	0	1	2	0

First, we need to read the module file. In the main part it was read line by line and copy the line into char array a [] from strcpy() function. Then we call the kmp function.

In kmp function first thing we did is make the sentence and the pattern to upper case because in problem that says solution should be case insensitive. So, after converting them to Upper case then we compare them. Then we need to calculate a prefix function π. That is done by the “PrefixFunction” function in the code. After that we compare pattern and the text. Comparing start from the index 0.If the letters of the pattern and the text matches, we increment both I and j. If the j value is equal to pattern length prints the line and increment count by 1. Then there’s a another if condition. In that if a mismatch found and j value not equal to 0, we go to our prefix array and check the pi value of the j-1 th letter.

And then go to that index and assign that value to the j. then again compare from that place. Else j is equal to 0 we increment I value by 1.