

A new pattern string search algorithm compared with other 4 pattern search algorithm:

1.
 - a. Search for files in the net with .txt extension (filetype:txt) then chose a txt file with more than five thousand words.
 - b. Find the character set of the text selected.
 - c. Build an array for each character set.
 - d. Record the index of each character occurrence in the text to be searched in each array of the corresponding character: you will build 26 arrays and an array for space. Ex:array for occurance of letter A.
2. Then do another net search for list of “internet search noise-words”.
Then build a table for the noise words of two or more characters. Search the text for the noise words and record the number of its occurrence in the table. Call this table noise words frequency table.
 - a. Use the arrays built in step 1.
 - b. Write the pseudo code of the algorithm
 - c. Code the algorithm and test it.
3. Do the test in step 2 using the algorithms of KMP, naïve search, Boyer-moore, rabin karp.

A search demo :

1-You have a text file containing 5000 words

2-build an array for each character to record where its location in the text
Its gonna be 26 arrays for each letter and an array for space

For example we are searching for the word “and” and we have to consider before and after the word ‘and’

space=[0,5,8,10,13,17]

a=[7,9,12,14,27,30,35]

n=[3,4,6,15,19,21,22]

d=[16,18,26,28,31,32]

3- first we look at space [0] then a should be located at a[1], but there is no one in array a
 4-then we test with space [5] we should have a[6] but no 6 in array a
 5-we look at space[8] and we find [9] in array a, so we should have a [10] in array n, but no such number exist in array n
 6-we return to space [10] but there is no [11] in array a
 7-we take space [13] and we have a[14] and we find n[15] and finally d[16] then space at [17] we found the pattern

Note:

- first you should search for all 25 noise words at once and calculate execution time and produce a table of the frequency of each word in the text file. Testing that with all 5 algorithms. And record the time execution and the frequency of each word in the log file
- second calculate execution time and frequency just for the entered noise word

The UI:

Search all 25 noise words at once	
Naïve search algoritm	Rabin-karp algorithm
Boyer-moore algorithm	KMP algorithm
New pattern search algorithm	
Execution time	<input type="text"/>
Search for a certain noise word	
Naïve search algoritm	Rabin-karp algorithm
Boyer-moore algorithm	KMP algorithm
New pattern search algorithm	
Execution time	Frequency
<input type="text"/>	<input type="text"/>

