CMPS 3120 Lab

Name Mihail Chitorog

1. Write a searching program by Horspool's method.

Input:	pattern (a string to search for)
	text (a long string to search in)
Output:	**Print the "Shift table"**
	How many matches can you find? Where are they located?
	How many comparisons did the program use?
Test Case:	"AAAAAAAAAAAAAAAAB" vs "AAAAB"
	"THIS IS A TEST TEXT" vs "TEST"
	"THIS IS A SIMPLE EXAMPLE" vs "EXAMPLE"

ALGORITHM *ShiftTable*(P[0..m− 1])
//Fills the shift table used by Horspool's and Boyer-Moore algorithms
//Input: Pattern P[0..m − 1] and an alphabet of possible characters
//Output: *Table*[0..size − 1] indexed by the alphabet's characters and
// filled with shift sizes computed by formula (7.1)
for i ← 0 **to** size − 1 **do** *Table*[i] ← m
for j ← 0 **to** m − 2 **do** *Table*[P[j]] ← m− 1− j
return *Table*

ALGORITHM *HorspoolMatching*(P[0..m− 1], T [0..n− 1])
//Implements Horspool's algorithm for string matching
//Input: Pattern P[0..m− 1] and text T [0..n− 1]
//Output: The index of the left end of the first matching substring
// or−1 if there are no matches
ShiftTable(P[0..m− 1]) //generate Table of shifts
i ← m − 1 //position of the pattern's right end
while i ≤ n − 1 **do**
    k ← 0 //number of matched characters
    while k ≤ m− 1 **and** P[m− 1− k]= T [i− k] **do**
        k ← k + 1
    if k= m
        return i− m + 1
    else i ← i + *Table*[T [i]]
return −1

2. Find out your previous lab for the Brute Force string matching program, try the above test cases and compare the difference .

Brute force string matching program results:

Running tests...

--- Test 1 ---

Pattern: 001011

Text: 10010101001001100101110010110 0

Matching positions: 15 22

Total number of comparisons: 55

--- Test 2 ---

Pattern: NOT

Text: NOBODY_NOTICED_HIM

Matching positions: 7

Total number of comparisons: 20

--- Test 3 ---

Pattern: N O T

Text: N O B O D Y _ N O T I C E D _ H I M

Matching positions: 14

Total number of comparisons: 39

Horspool's string matching program results:

Running tests...

--- Test 1 ---

==========

Text: "10010101001001100101110010 1100"
Pattern: "001011"
==========

Shift Table:

'0': 2
'1': 1
Default: 6

Number of matches found: 2
Matches located at positions: 15 22
Total number of character comparisons: 37

--- Test 2 ---

==========

Text: "NOBODY_NOTICED_HIM"
Pattern: "NOT"
==========

Shift Table:

'N': 2
'O': 1
'T': 3
Default: 3

Number of matches found: 1
Matches located at positions: 7
Total number of character comparisons: 8

--- Test 3 ---

==========

Text: "N O B O D Y _ N O T I C E D _ H I M"
Pattern: "N O T"
==========

Shift Table:

'N': 4
' ': 1
'O': 2
'T': 5
Default: 5

Number of matches found: 1
Matches located at positions: 14
Total number of character comparisons: 14

Brute force takes more comparisons than Horspool's, hence brute force is slower.