**Pseudo Code:**

S*hiftTable*(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*

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] )

i ← m – 1 //generate Table of shifts  
**while** i ≤ n − 1 **do**  //position of the pattern’s right end

k ← 0  
**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

**Complexity**

For Horspool its Θ(n) and all that algorithm is called within a for loop so it becomes Θ(n2)