

Sequence algorithms

by Samuel Ortion

Prof.: Fariza Tahi

2024

Contents

1	Back to basics	3
2	Motif	5

1 Back to basics

Algorithm 1 Search an element in an array

```
1: function Search(A: Array(n), E: element)
2:   for (i=0; i < n; i++) do
3:     if A[i] = E then
4:       return true
5:     end if
6:   end for
7:   return false
8: end function
```

Algorithm 2 Search an element in an array using a while loop

```
1: function Search(A: Array(n), E: element)
2:    $i \leftarrow 0$ 
3:   while  $i < n$  do
4:     if A[i] = E then
5:       return true
6:     end if
7:      $i \leftarrow i + 1$ 
8:   end while
9:   return false
10: end function
```

Algorithm 3 Search an element in an array using a while loop (bis)

```

1: function Search(A: Array( $n$ ), E: element)  $\triangleright$  Version preferred by the professor
2:    $i \leftarrow 0$ 
3:   while  $i < n$  and  $A[i] \neq E$  do
4:     if  $A[i] = E$  then
5:       return true
6:     end if
7:      $i \leftarrow i + 1$ 
8:   end while
9:   if  $i = n$  then
10:    return false
11:  else
12:    return true
13:  end if
14: end function

```

Algorithm 4 Count the occurrences of an element in an array

```

1: function Search(A: Array( $n$ ), E: element)
2:    $c \leftarrow 0$ 
3:   for  $i = 0; i < n; i++$  do
4:     if  $A[i] = E$  then
5:        $c \leftarrow c + 1$ 
6:     end if
7:   end for
8:   return  $c$ 
9: end function

```

2 Motif

Algorithm 5 Brute-force search of a motif in a sequence

```
1: function FindMotif(S: Array( $n$ ), M: Array( $m$ ))
2:   Returns list of position
3:    $pos \leftarrow$  empty list
4:    $i \leftarrow 0$ 
5:   while  $i < n - m + 1$  do
6:      $j \leftarrow 0$ 
7:     while  $j < m$  and  $S[i+j] = M[j]$  do
8:        $j++$ 
9:     end while
10:    if  $j = m$  then
11:      add  $i$  to  $pos$ 
12:    end if
13:     $i++$ 
14:  end while
15:  return  $pos$ 
16: end function
```

Algorithm 6 Knuth-Morris-Pratt algorithm

```

1: function KMP_Search(S: Array(n), M: Array(m))
2:   Returns Integer
3:    $table \leftarrow \text{KMP\_Table}(M)$ 
4:    $c \leftarrow 0$   $\triangleright$  Count the number of matches
5:    $i \leftarrow 0$ 
6:    $j \leftarrow 0$ 
7:   while  $i < n$  do
8:     if  $S[i] = M[j]$  then
9:        $i \leftarrow i + 1$ 
10:       $j \leftarrow j + 1$ 
11:    end if
12:    if  $j = m$  then
13:       $c \leftarrow c + 1$ 
14:       $j \leftarrow table[j - 1]$ 
15:    else if  $j < n$  and  $M[j] \neq S[i]$  then
16:      if  $j \neq 0$  then
17:         $j \leftarrow table[j - 1]$ 
18:      else
19:         $i \leftarrow i + 1$ 
20:      end if
21:    end if
22:  end while
23:  return  $c$ 
24: end function
25: function KMP_Table(M: Array(m))
26:   Returns Array(m)
27:    $previous \leftarrow 0$ 
28:    $table \leftarrow$  array of zeros of size  $m$ 
29:   for  $i = 0; i < m; i++$  do
30:     if  $M[i] = M[previous]$  then
31:        $previous \leftarrow previous + 1$ 
32:        $table[i] \leftarrow previous$ 
33:      $i \leftarrow i + 1$ 
34:   else
35:     if  $previous = 0$  then
36:        $previous \leftarrow table[previous - 1]$ 
37:     else
38:        $table[i] \leftarrow 0$ 
39:        $i \leftarrow 1$ 
40:     end if
41:   end if
42: end for
43: end function

```

3