GENIOMHE

Sequence algorithms

by Samuel Ortion

Prof.: Fariza Tahi

2024

Contents

1 Back to basics

Algorithm 1 Search an element in an array

```
function Search(A: Array(n), E: element)
    for (i=0; i < n; i++) do
        if A[i] = E then
            return true
        end if
    end for
    return false
end function
```

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

```
function Search(A: Array(n), E: element)
    $i \leftarrow 0$
    while $i < n$ do
        if A[i] = E then
            return true
        end if
        $i \leftarrow i + 1$
    end while
    return false
end function
```

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

```
function Search(A: Array(n), E: element)        ▷ Version “preffered” by the professor
    $i \leftarrow 0$
    while $i < n$ and $A[i] \neq E$ do
        if A[i] = E then
            return true
        end if
        $i \leftarrow i + 1$
    end while
    if $i = n$ then
        return false
    else
        return true
    end if
end function
```

Algorithm 4 Count the occurences of an element in an array

```
function Search(A: Array(n), E: element)
    $c \leftarrow 0$
    for $i = 0; i < n; i + +$ do
        if A[i] = E then
            $c \leftarrow c + 1$
        end if
    end for
    return $c$
end function
```

2 Motif

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

```
function FindMotif(S: Array(n), M: Array(m))
    Returns list of position
    $pos \leftarrow$ empty list
    $i \leftarrow 0$
    while $i < n - m + 1$ do
        $j \leftarrow 0$
        while $j < m$ and S[i+j] = M[j] do
            $j++$
        end while
        if $j = m$ then
            add $i$ to $pos$
        end if
        $i++$
    end while
    return $pos$
end function
```

Algorithm 6 Knuth-Morris-Pratt algorithm

```
function KMP_Search(S: Array(n), M: Array(m))
    Returns Integer
    $table \leftarrow$ KMP_Table($M$)
    $c \leftarrow 0$                                        ▷ Count the number of matches
    $i \leftarrow 0$
    $j \leftarrow 0$
    while $i < n$ do
        if $S[i] = M[i]$ then
            $i \leftarrow i + 1$
            $j \leftarrow j + 1$
        end if
        if $j = m$ then
            $c \leftarrow c + 1$
            $j \leftarrow table[j - 1]$
        else if $j < n$ and $M[j] \neq S[i]$ then
            if $j \neq 0$ then
                $j \leftarrow table[j - 1]$
            else
                $i \leftarrow i + 1$
            end if
        end if
    end while
    return $c$
end function
function KMP_Table(M: Array(m))
    Returns Array(m)
    $previous \leftarrow 0$
    $table \leftarrow$ array of zeros of size m
    for $i = 0; i < m; i++$ do
        if $M[i] = M[previous]$ then
            $previous \leftarrow previous + 1$
            $table[i] \leftarrow previous$
            $i \leftarrow i + 1$
        else
            if $previous = 0$ then
                $previous \leftarrow table[previous - 1]$
            else
                $table[i] \leftarrow 0$
                $i \leftarrow 1$
            end if
        end if
    end for
end function
```