Search's algorithm of Pisma

Pisma search's algorithm function

Summary of the "Search" function

The first thing this function does is read the file in which the information to be processed is stored. The file name is stored in the global variable "nomArch". The buffer that is associated with the reading of the file is stored in the variable "input". Events will only be executed if the file exists. In any case this does not exist, an error message will be sent to the console. The first thing that is done in this process is to get the number of characters the file contains. This is stored in the variable "max". Also, store the values of the file inside the "letter" vector. The first of the three processes is based on this search algorithm, it is in charge of eliminating all the characters that should not appear. Each character contained in "letter" will be copied against each of the four possibilities (A, T, G and C). In the case that a match is not confirmed, this character is removed from the list. In the diagram (1) the generalized behavior of this one is shown. This makes use of a function called 'Mayuscula' that transforms the character that receives in its corresponding upper case.

The second part of the algorithm is responsible for separating all the characters that were received in strings of 100 characters. For this a '\ n' character is applied every 100 values. A character vector is then converted to a single word using the toString function (whose Java equivalent is copyValueOf). An evaluation of the phrase is also performed in order for the character '\ n' to be used as a line break. This is in charge 'Evaluate' command (whose equivalence in Java is trim). In the diagram (2) the generalized behavior of this one is shown.

The third part that makes up this function searches for the pattern contained in the vector 'string'. To do this, it looks for the first value to match any of the characters of the word contained in 'letter'. In case of coincidence it is continued with the search of the rest of the values. In the diagram (3) the generalized behavior of this one is shown.

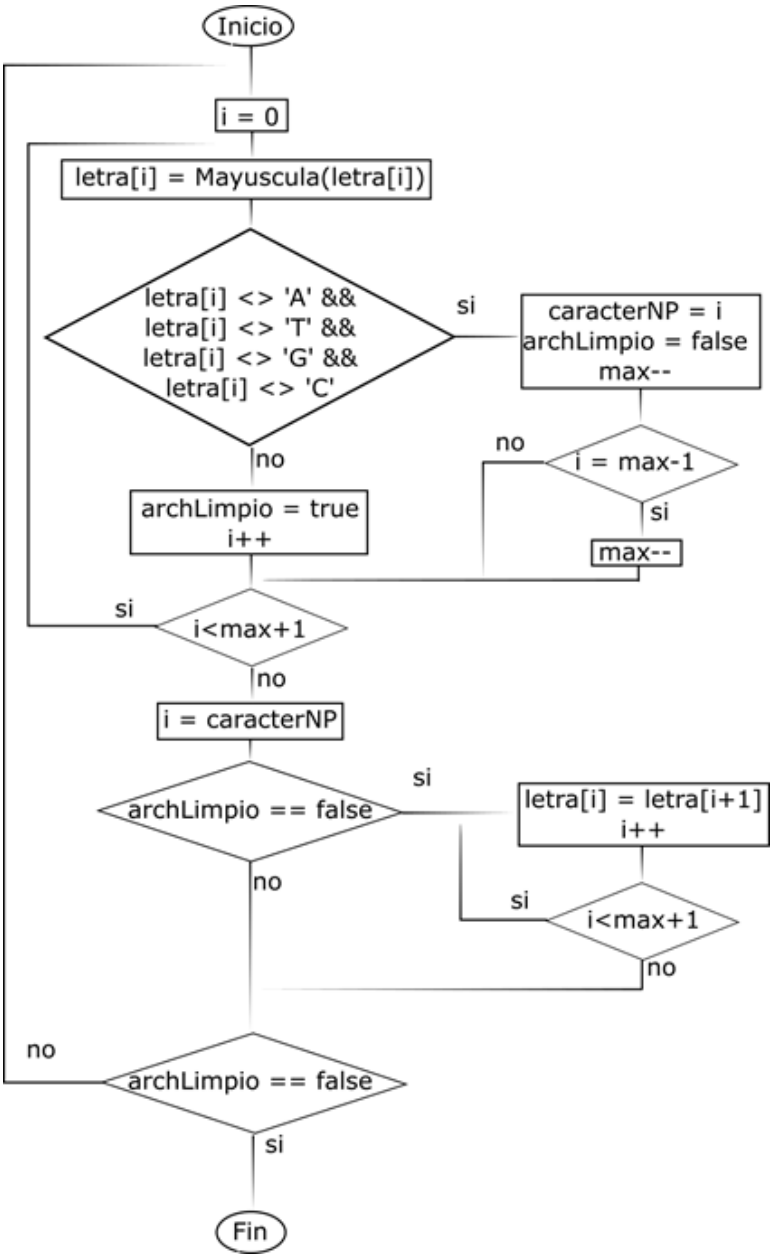
Inicio
i = 0
letra[i] = Mayuscula(letra[i])
letra[i] <> 'A' &&
letra[i] <> 'T' &&
letra[i] <> 'G' &&
letra[i] <> 'C'
si
caracterNP = i
archLimpio = false
max--
no
archLimpio = true
i++
no
i = max-1
si
max--
si
i<max+1
no
i = caracterNP
archLimpio == false
si
letra[i] = letra[i+1]
i++
no
si
i<max+1
no
no
archLimpio == false
si
Fin

diagram (1)

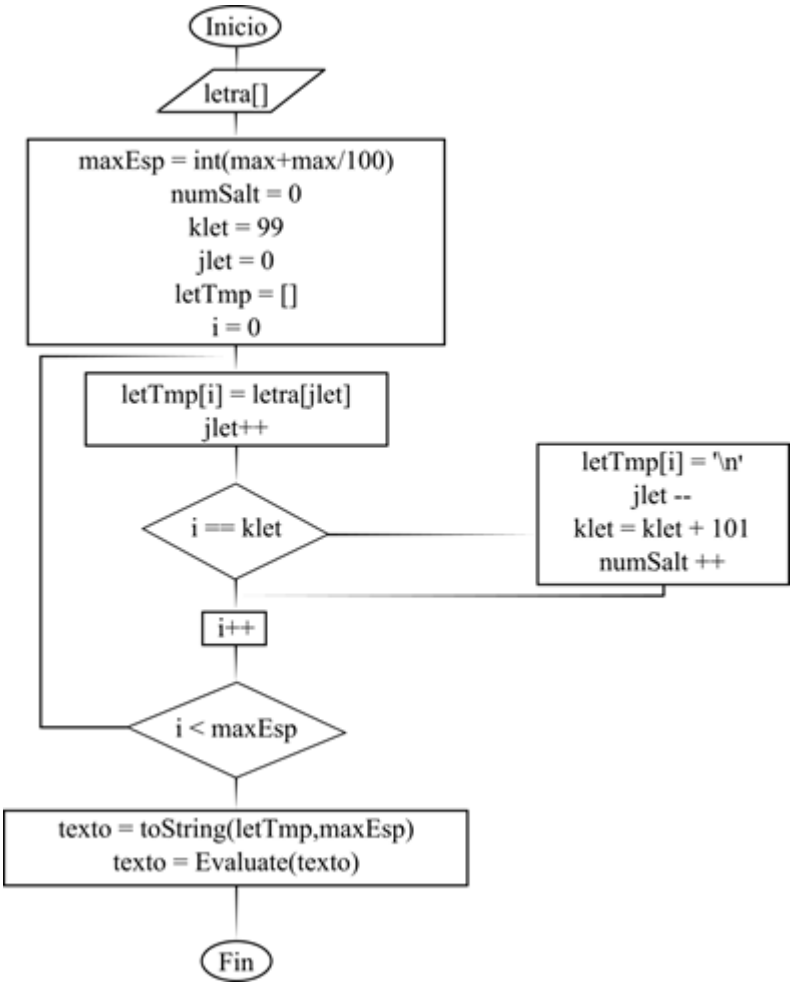
Inicio
letra[]
maxEsp = int(max+max/100)
numSalt = 0
klet = 99
jlet = 0
letTmp = []
i = 0
letTmp[i] = letra[jlet]
jlet++
i == klet
letTmp[i] = '\n'
jlet --
klet = klet + 101
numSalt ++
i++
i < maxEsp
texto = toString(letTmp,maxEsp)
texto = Evaluate(texto)
Fin

diagram (2)

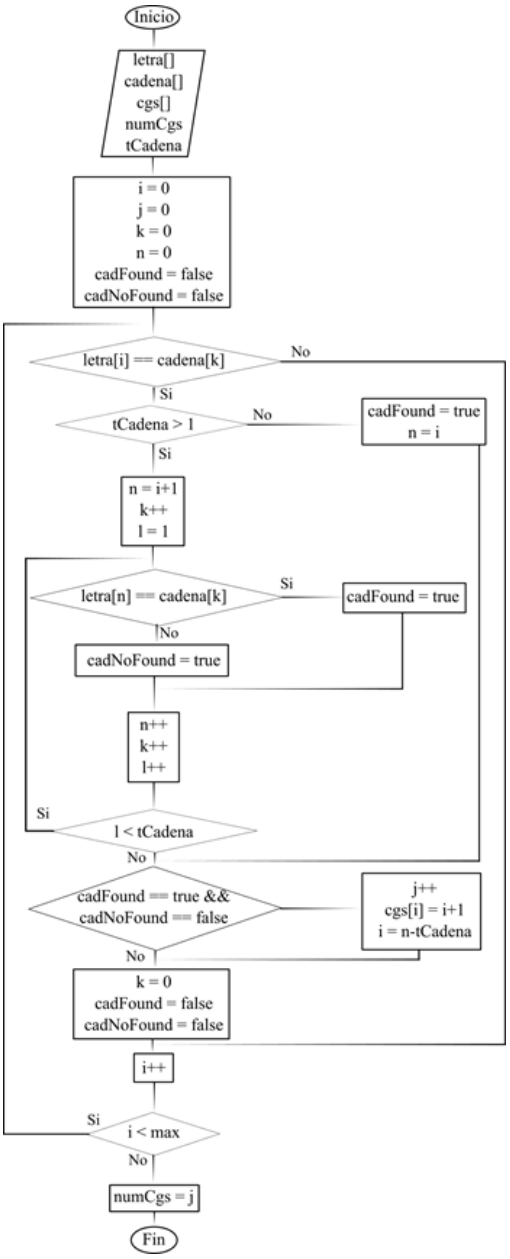
Inicio
letra[]
cadena[]
cgs[]
numCgs
tCadena
i = 0
j = 0
k = 0
n = 0
cadFound = false
cadNoFound = false
letra[i] == cadena[k]
No
Si
tCadena > 1
No
cadFound = true
n = i
Si
n = i+1
k++
l = 1
letra[n] == cadena[k]
Si
cadFound = true
No
cadNoFound = true
n++
k++
l++
Si
l < tCadena
No
cadFound == true &&
cadNoFound == false
j++
cgs[i] = i+1
i = n-tCadena
No
k = 0
cadFound = false
cadNoFound = false
i++
Si
i < max
No
numCgs = j
Fin

diagram (3)

Search's function pseudo code

Global variables

```
Integer:      numCgs
              tCadena
              cgs[]
Character:    letra[]
              cadena[]
String:       nomArchivo
```

Local variables

```
Integer:      caracterNP = 0
              i = 0
              j = 0
              k = 0
              n = 0
              max = 0
              maxEsp = 0
              klet = 99
              jlet = 0
Boolean:      archLimpio = false
              cadFound = false
              cadNoFound = false
File:         F = null
Character:    letTmp[]
```

Algorithm

```
F = OpenFile(nomArchivo)
If exists(F) Then
        letra[] = ReadInformation(F)
        While archivoLimpio <> false Do
                max = FileLength(F)
                i = 0
                While i < max+1 Do
                        letra[i] = CapitalLetter(letra[i])
                        If letra[i] <> 'A' && letra[i] <> 'T' && letra[i] <> 'G' && letra[i] <> 'C' Then
                                caracterNP = i
                                archivoLimpio = false
                                max = max - 1
                                If i == max-1 Then
                                        max = max – 1
                                EndIf
                        Else
                                archLimpio = true
                                i = i + 1
                        EndIf
                EndWhile
                i = caracterNP
                If archivoLimpio == false Then
                        While i < max+1 Do
```

```
                letra[i] = letra[i+1]
                i = i + 1
            EndWhile
        EndIf
EndWhile
maxEsp = BecomeInteger(max+max/100)
letTmp[] = VectorCharacter(maxEsp)
i = 0
While i < maxEsp Do
        letTmp[i] = letra[jlet]
        jlet = jlet + 1
        If i == klet Then
            letTmp[i] = '\n'
            jlet = jlet – 1
            klet = klet + 101
            numSalt = numSalt + 1
        EndIf
        i = i + 1
EndWhile
texto = BecomeString(letTmp,maxEsp)
texto = Evaluate(texto)
i = 0
While i < max Then
        If letra[i] == cadena[k] Then
            If tCadena > 1 Then
                n = i + 1
                k = k + 1
                l = 1
                While l < tCadena Do
                    If letra[n] == cadena[k] Then
                        cadFound = true
                    Else
                        cadNoFound = true
                    EndIf
                    n = n + 1
                    k = k + 1
                    l = l + 1
                EndWhile
            Else
                cadFound = true
                n = i
            EndIf
            If cadFound == true && cadNoFound == false Then
                j = j + 1
                cgs[i] = i + 1
                i = n – tCadena
            EndIf
            k = 0
            cadFound = false
            cadNoFound = false
        EndIf
        i = i + 1
EndWhile
```

```
        numCgs = j
        CerrarArchivo(F)
EndIf
```