

Proyecto - Entrega 1

Domínguez Aspilcueta, Pedro Francisco - 201910375

Lama Carrasco, Miguel Angel - 201910199

Ríos Vásquez, Paul Jeremy - 201910038



Universidad de Ingeniería y Tecnología

Ciencia de la computación

Análisis y Diseño de Algoritmos 1.00

Docente: Gutierrez Alva, Juan Gabriel

TA: Lopez Condori, Rodrigo

Pregunta 2

RECIBE: a_1 de una cadena no vacía $s = \{a_1 a_2 \cdots a_n\}$

DEVUELVE: Una cadena $r = \{f_k(a_1)f_k(a_2) \cdots f_k(a_n)\}$ tal que $1 \leq k \leq 4$ y r es de tamaño mínimo óptimo

```
    OPT( $i, j$ )
1: if ( $a_i \in C_j$ ) then
2:      $r = r + f_j(a_i)$ 
3: else
4:      $r = r + \dots$ 
5: if  $i = i_{ultimo}$  then
6:     return  $r$ 
7:  $t_j = \dots$ 
8:  $t_{(j+1) \% 4} = T_{(j+1) \% 4}$ 
9:  $t_{(j+2) \% 4} = T_{(j+2) \% 4}$ 
10:  $t_{(j+3) \% 4} = T_{(j+3) \% 4}$ 
11:  $i = i + 1$ 
12:  $r = r + \min(r + t_1 + OPT(i, 1), r + t_2 + OPT(i, 2), r + t_3 + OPT(i, 3), r +$ 
     $t_4 + OPT(i, 4))$ 
13: return  $r$ 
```

En la primera recurrencia se calcula la codificación del caracter s_0 para posteriormente concatenarlo con la mínima longitud de las 4 posibles codificaciones C_1, C_2, C_3, C_4 para s_1 .

Esta recurrencia se realiza para toda la cadena s hasta llegar al caso base s_n donde se retorna $f_n(a_n)$ siendo así que se calcula todas las posibles combinaciones y se garantiza que se obtiene la solución óptima.

Complejidad

Si existen C_1, C_2, C_3, C_4 y se calculan todas las posibles codificaciones para todo caracter de s . Por lo tanto, el algoritmo tiene una complejidad de $O(4^n)$ donde n es la longitud de la cadena.