

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

- Sea s una cadena no vacía tal que $s = \{a_1 a_2 \cdots a_n\}$.
- $\{C_0, C_1, C_2, C_3\}$ los modos de codificar a .
- B una cadena de dos bits $\{00, 01, 10, 11\}$ que indica cual C comienza una codificación de s .
- $\{T_0, T_1, T_2, T_3\}$ las cadenas de transición a otro C . Tomando en cuenta que T puede ser vacío si no se requiere ninguna transición.
- r una codificación de s tal que $r = \{B \cdot C(a_1) \cdot T \cdot C(a_2) \cdot T \cdot C(a_3) T \cdots T \cdot C(a_n)\}$.

1. Pregunta 2

$$OPT(i, j) \begin{cases} B_j + C_j(a_i) & i = 1; a_i \in C_j \\ \infty & a_i \notin C_j \\ \min\{OPT(i-1, j) + C_j(a_i), \\ OPT(i-1, (j+1) \% 4) + T_{(j+1) \% 4} + C_j(a_i), \\ OPT(i-1, (j+2) \% 4) + T_{(j+2) \% 4} + C_j(a_i), \\ OPT(i-1, (j+3) \% 4) + T_{(j+3) \% 4} + C_j(a_i)\} & i > 1 \end{cases}$$

2. Pregunta 3

RECIBE: un índice final i de una cadena s y un índice j de C .

DEVUELVE: Una cadena r que codifica a s de manera mínima óptima con $C_j(a_i)$.

```

    OPT(i, j)
1: if ( $a_i \in C_j$ ) and  $i = 1$  then
2:   return  $B_j + C_j(a_i)$ 
3: else if ( $a_i \notin C_j$ ) then
4:   return  $\infty$ 
5:  $A_0 = C_j(a_i)$ 
6:  $A_1 = T_{(j+1) \% 4} + C_j(a_i)$ 
7:  $A_2 = T_{(j+2) \% 4} + C_j(a_i)$ 
8:  $A_3 = T_{(j+3) \% 4} + C_j(a_i)$ 
9:  $r = \min(OPT(i-1, j) + A_0, OPT(i-1, (j+1) \% 4) + A_1, OPT(i-1, (j+2) \% 4) + A_2, OPT(i-1, (j+3) \% 4) + A_3)$ 
10: return  $r$ 

```

RECIBE: una cadena s no vacía.

DEVUELVE: Una cadena que codifica a s de manera mínima óptima.

MIN-COD(s)

1: Sea a_n el caracter final de s .

2: **return** $\min(OPT(a_n, 0), OPT(a_n, 1), OPT(a_n, 2), OPT(a_n, 3))$

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.

2.1. Pregunta 5

RECIBE: una cadena s .

DEVUELVE: Una cadena r que codifica a s de manera mínima óptima.

MIN-COD-DIN(s)

1: Sea A una matriz que guarda la solución.

2: $B = \{00, 01, 10, 11\}$ tal que indica cual es el modo de codificación inicial.

3: **for** $j=1$ to 4 **do**

4: $A[0][j] = B_j$

5: **for** $i = 1$ to n **do**

6: $A[i][1] = \infty$

7: $A[i][2] = \infty$

8: $A[i][3] = \infty$

9: $A[i][4] = \infty$

10: **for** $j = 1$ to 4 **do**

11: **if** Si a_i puede ser codificado con C_j **then**

12: $M_1 = A[i-1][j] + C_j(a_i)$

13: $M_2 = A[i-1][(j+1) \% 4] + T_{(j+1) \% 4} + C_j(a_i)$

14: $M_3 = A[i-1][(j+2) \% 4] + T_{(j+2) \% 4} + C_j(a_i)$

15: $M_4 = A[i-1][(j+3) \% 4] + T_{(j+3) \% 4} + C_j(a_i)$

16: $A[i][j] = \min(M_1, M_2, M_3, M_4)$

17:

18: $r = \min(A[n][1], A[n][2], A[n][3], A[n][4])$

19: **return** r