



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACION

Diseño y Análisis de Algoritmos - IIC2283

Interrogación 2

Tiempo: 2.5 horas

1. [1 punto] El siguiente es el código para el procedimiento **CalcularCodificaciónHuffman** analizado en clases:

CalcularCodificaciónHuffman(f)

Sea Σ el dominio de la función f

if $\Sigma = \{a, b\}$ **then return** $\{(a, 0), (b, 1)\}$

else

Sean $a, b \in \Sigma$ tales que $a \neq b$, $f(a) \leq f(b)$ y

$f(b) \leq f(e)$ para todo $e \in (\Sigma \setminus \{a, b\})$

Sea c un símbolo que **no** aparece en Σ

$g := (f \setminus \{(a, f(a)), (b, f(b))\}) \cup \{(c, f(a) + f(b))\}$

$\tau^* := \text{CalcularCodificaciónHuffman}(g)$

$w := \tau^*(c)$

$\tau := (\tau^* \setminus \{(c, w)\}) \cup \{(a, w0), (b, w1)\}$

return τ

Demuestre que si f es una función de frecuencias relativas, Σ es el dominio de f y $\tau = \text{CalcularCodificaciónHuffman}(f)$, entonces τ es una Σ -codificación libre de prefijos que minimiza la función $\text{lp}_f(x)$.¹

2. [1.5 puntos] Recuerde que decimos que una función $f : \mathbb{N} \rightarrow \mathbb{N}$ es estrictamente creciente si para todo $n \in \mathbb{N}$ se tiene que $f(n) < f(n+1)$.

Construya un algoritmo que reciba como entrada un par (f, ℓ) , donde $f : \mathbb{N} \rightarrow \mathbb{N}$ es una función estrictamente creciente y $\ell \in \mathbb{N}$, y retorne el valor:

$$m = \min \{i \in \mathbb{N} \mid f(i) > \ell\}$$

Analice la complejidad de su algoritmo considerando la evaluación de la función f como la operación básica, y suponiendo que calcular $f(i)$ tiene costo 1 para cada $i \in \mathbb{N}$.

Importante: El algoritmo desarrollado en esta pregunta debe funcionar en tiempo $O(\log_2(m))$, donde m es el valor retornado por el algoritmo dada la entrada (f, ℓ) .

3. [1.5 puntos] Toda palabra w puede ser escrita como la concatenación de una lista de palíndromos $[w_1, w_2, \dots, w_k]$, vale decir, $w = w_1 w_2 \dots w_k$ donde cada w_i es un palíndromo. Por ejemplo, la palabra $bcbcb$ puede ser escrita como la concatenación de las siguientes listas de palíndromos: $[b, bcb]$, $[b, c, b, c]$ y $[bcb, c]$.

¹Recuerde que $\text{lp}_f(\tau) = \sum_{a \in \Sigma} f(a) \cdot |\tau(a)|$

Dada una palabra w , defina $\text{MinPal}(w)$ como el menor largo de las listas $[w_1, w_2, \dots, w_k]$ de palíndromos tales que $w = w_1 w_2 \dots w_k$. Por ejemplo, tenemos que $\text{MinPal}(bcbc) = 2$, mientras que $\text{MinPal}(bcbcb) = 1$ puesto que $bcbcb$ es un palíndromo (y por lo tanto, $bcbcb$ puede ser escrita como la concatenación de la lista de palíndromos $[bcbcb]$).

Utilizando la técnica de programación dinámica construya un algoritmo que, dada una palabra w , calcule $\text{MinPal}(w)$. Analice la complejidad de su algoritmo y muestre que funciona en tiempo $O(|w|^k)$ para alguna constante k .

4. Resuelva los siguientes problemas.

- (a) [1 punto] Para reducir el valor de un número natural $n > 1$ consideramos dos operaciones: restar 1 y dividir por 2 si el número n es par, las cuales son denotadas como -1 y $/2$, respectivamente. Una secuencia de estas operaciones se dice completa para n si el resultado de aplicarla a partir de n es el número 1. Por ejemplo, las siguientes son secuencias completas de las operaciones -1 y $/2$ para el número 9:

$$\begin{array}{cccccccccccc} 9 & \xrightarrow{-1} & 8 & \xrightarrow{-1} & 7 & \xrightarrow{-1} & 6 & \xrightarrow{/2} & 3 & \xrightarrow{-1} & 2 & \xrightarrow{-1} & 1 \\ 9 & \xrightarrow{-1} & 8 & \xrightarrow{/2} & 4 & \xrightarrow{-1} & 3 & \xrightarrow{-1} & 2 & \xrightarrow{/2} & 1 & & \end{array}$$

Construya un algoritmo codicioso que dado $n > 1$ retorne una secuencia de operaciones -1 y $/2$ completa para n y de largo mínimo. Explique por qué su algoritmo es codicioso y demuestre que es correcto. Además, analice la complejidad de su algoritmo y muestre que funciona en tiempo polinomial en el tamaño de la entrada.

- (b) [1 punto] Para aumentar el valor de un número natural consideramos dos operaciones: sumar 1 y multiplicar por 2, las cuales son denotadas como $+1$ y $\times 2$, respectivamente. Una secuencia de estas operaciones se dice completa para un número natural $n > 1$ si el resultado de aplicarla a partir de 1 es el número n . Por ejemplo, las siguientes son secuencias completas de las operaciones $+1$ y $\times 2$ para el número 8:

$$\begin{array}{cccccccc} 1 & \xrightarrow{+1} & 2 & \xrightarrow{+1} & 3 & \xrightarrow{+1} & 4 & \xrightarrow{\times 2} & 8 \\ 1 & \xrightarrow{\times 2} & 2 & \xrightarrow{\times 2} & 4 & \xrightarrow{+1} & 5 & \xrightarrow{+1} & 6 & \xrightarrow{+1} & 7 & \xrightarrow{+1} & 8 \end{array}$$

Construya un algoritmo codicioso que dado $n > 1$ retorne una secuencia de operaciones $+1$ y $\times 2$ completa para n y de largo mínimo. Explique por qué su algoritmo es codicioso y demuestre que es correcto. Además, analice la complejidad de su algoritmo y muestre que funciona en tiempo polinomial en el tamaño de la entrada.