Huffman Code

jueves, 9 de noviembre de 2023 09:04

- Σ alfabeto de caracteres $\{a, b, c, d, e, f\}$
- F archivo que contiene 100 k caracteres.
- Conocemos la frecuencia que aparecen los caracteres en F

	a	b	c	d	e	f
f (miles)	45	13	12	16	9	5

- Queremos almacenar F

Objetivo, usar el menor espacio posible.

Tools:

Código binario de Caracteres: (o código)

$$\mathcal{C} : \Sigma \longrightarrow \{0,1\}^+ \quad \text{inyectiva}$$
$$\text{``}x\text{''} \longmapsto \mathcal{C}(x)$$

010 000 0111 1 10 010

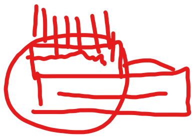
Decimos que $\mathcal{C}(x)$ es el código del caracter "x".

	a	b	c	d	e	f
Frequency (in thousands)	45	13	12	16	9	5
Fixed-length codeword	000	001	010	011	100	101
Variable-length codeword	0	101	100	111	1101	1100

0 10 11 010 000 0111 00 1111

00001 cab acb aaab

x y



C(x) C(y)

Código libre de prefijos

Códigos donde $x \neq y \Rightarrow C(x)$ no es prefijo de $C(y)$

Problema: Obtener

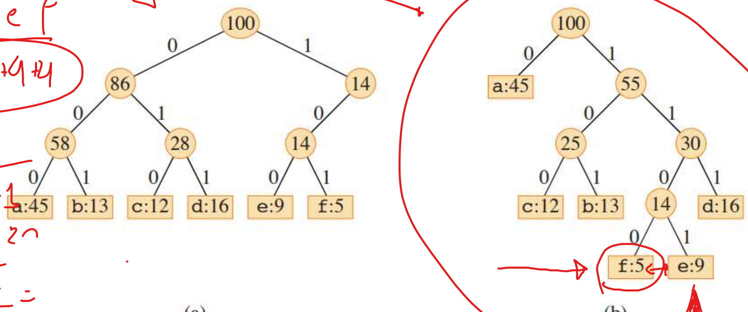
$$\min \left\{ \sum_{x \in F} |\mathcal{C}(x)| : \mathcal{C} \text{ código libre de prefijos de } \Sigma \right\}.$$

Prefix-free code ⟷ Binary Tree, leaves with characters

abbcddef

1+3·3+3·3+3+4+4

	f	d
a	1	1
b	2	3
c	1	3
d	2	3
e	1	4
f	1	4



(a) (b)

$|c(b)|$ → 3

T

Prop: optimo se alcanza con un full binary tree.

Propiedad: $d_T : \Sigma \longrightarrow \mathbb{Z}_0^+$

$x \longmapsto d_T(x)$ propundidad

$$val(T) = \sum_{x \in \Sigma} d_T(x) \cdot f(x)$$

Prob: Obtener min:

$\{ val(T) : T$ full binary tree associated to a pfc C de $\Sigma \}$

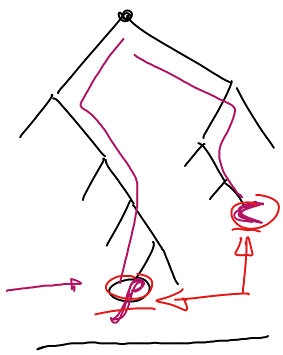
Lema 1: Sea $c \in \Sigma$ el caracter con menor frecuencia

$\Rightarrow$ Existe un T optimo Tq c se ubica en un nodo con propundidad maxima

- Sea T un óptimo:
  Caso 1: c Tiene prof maxima. Trivial ✓
  Caso 2: c no Tiene prof máxima.

$\Rightarrow \exists\, p \in \Sigma$ con prof maxima → $d_T(p) > d_T(c)$

T' (c)



Construimos T' swapeando c con p.

Veamos que

$$d_{T'}(x) = \begin{cases} d_T(x) & x \notin \{c, p\} \\ d_T(p) & x = c \\ d_T(c) & x = p \end{cases}$$

Comparando T y T':

$$val(T) - val(T') = \sum_{x\in\Sigma} f(x)\cdot d_T(x) - \sum_{x\in\Sigma} f(x)\cdot d_{T'}(x).$$

$$= f(c)\cdot d_T(c) + f(p)\cdot d_T(p) - f(c)\cdot d_{T'}(c) - f(p)\, d_{T'}(p)$$

$$= f(c)\cdot \overset{(\le 0)}{d_T}(c) + f(p)\cdot \overset{(<0)}{d_T}(p) - f(c)\cdot d_T(p) - f(p)\, d_T(c)$$

$$= \underbrace{[f(c) - f(p)]}_{-}\ \underbrace{[d_T(c) - d_T(p)]}_{(-)} \geq 0.$$

$val(T) \geq val(T')$. $\longrightarrow$ T' es un arbol optimo

Lema 2: • Sean b y c los caracteres con menor frecuencia $(f(c) \le f(b))$

$\Rightarrow$ Existe un T optimo tq b y c son hermanos y tengan profundidad maxima

Prueba

Por Lema 1, existe T óptimo tq c tiene profundidad máxima

Como T es lleno, c tiene un hermano: x.

Casos: $x = b$ Trivial!

Caso 2: $x \neq b$. Aquí tenemos $d_T(b) \leq d_T(x)$ y $f(x) \geq f(b)$

T' swapeando x y b. Por el mismo proceso de Lema 1 ~~vez~~ se demuestra que T' también es optima.

Problema: • Σ alfabeto : $|\Sigma| = n$
• $f_n : \Sigma \to \mathbb{Z}_0^+$ frecuencia

Sol: Arboles binarios llenos con n hojas, cada una asociada a un caracter de Σ

$$val(T) = \sum_{x\in\Sigma} f(x)\cdot d_T(x)$$

Sol optima: Aquel T que minimiza $val(T)$.

Elección greedy: Escojo los 2 caracteres con menor frecuencia (b y c)

Subproblema:
- $\Sigma' = \Sigma - \{b, c\} \cup \{bc\}$
- $f_{n-1} : \Sigma' \to \mathbb{Z}^+$

$$f_{n-1}(x) = \begin{cases} f_n(b) + f_n(c) & x = bc \\ f_n(x) & x \neq bc \end{cases}$$

elección Greedy:
Existe un T optimo tq tiene a b y c en la mayor propundidad.

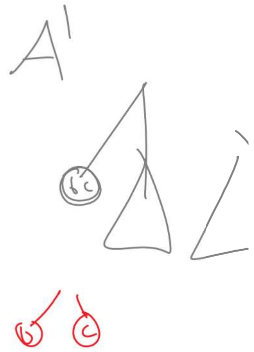
Prueba: Lema 2

Sub problema:
Dado T que tiene a b y c en la mayor profundidad.
y T' es el arbol dejado

$\Rightarrow$ Pg T' es optimo del problema P'!

Por contradicción, sup q T' no es optimo

$\Rightarrow \exists A'$ optimo de P'! $\quad val(A') < val(T')$



Construimos A:
agarramos A', ubicamos la hoja asociada a bc,
y le damos 2 hijos: b y c.

Claramente A es sol de P (~~arb bin lleno n hojas ... Σ~~)

$$val(A) = val(A') - f_{n-1}(bc) \cdot d_{A'}(bc) + f_n(b) \cdot d_A(b) + f_n(c)\, d_A(c)$$

$$val(A) - val(A') = -[f(b) + f(c)] \cdot x + f_n(b)\,(x+1) + f_n(c) \cdot (x+1)$$

$$= f_n(b) + f_n(c) = f_{n-1}(bc)$$

Similarmente

$$val(T) - val(T') = f_n(b) + f_n(b),$$

$$\Rightarrow val(A) - \underbrace{val(A')}_{} = val(T) - val(T') -$$

$$- val(A) = val(A) - val(A') + val(A') = val(T) - val(T') + val(A') \Rightarrow \Leftarrow$$

$$< val(T)$$