Laboratori 3 CDI-FIB

Curs 2022/2023

12 de març de 2023

Entropia. Definiu una funció `entropy(txt,k=1,pre="")` que calculi l'entropia d'una font creada a partir d'un text `txt` (un string de Python), on els paràmetres indiquen:

- `k` és un enter $\geqslant 1$ que dona la longitud dels blocs de caràcters del text que es consideren les lletres de la font; per defecte `k` val 1 i per tant les lletres de l'alfabet de la font són lletres de les que apareixen al text.
- `pre` és un string de Python que diu que les lletres de la font són només els blocs de `k` lletres que en el text `txt` venen a continuació de `pre`; per defecte aquest prefix és la paraula buida i en aquest cas s'agafen tots els blocs de longitud `r`.

Longitud mitjana. Escriviu una funció `mean_length(src,code)` que retorni la longitud mitjana d'un codi de font. El primer paràmetre és una font d'informació de la forma

$$\texttt{src} = [(\texttt{a1}, \texttt{w1}), (\texttt{a2}, \texttt{w2}), \ldots, (\texttt{an}, \texttt{wn})]$$

on els parells `(ai,wi)` contenen les lletres `ai` (o paraules d'un codi prefix) emeses per la font i els pesos `wi` (enters $\geqslant 1$) indiquen les freqüències amb què s'emet cadascuna: les probabilitats són $\frac{\texttt{wi}}{\texttt{w}}$ amb `w` la suma de les `wi`.

El segon paràmetre és un codi `codi=[c1,c2,...,cn]` que conté tantes paraules com lletres emet la font, de manera que la codificació de cada lletra `ai` és la paraula codi `ci`.

Codi de Huffman. Escriviu una funció `huffman_code(src)` amb paràmetre una font, que com abans és una llista

$$\texttt{src} = [(\texttt{a1}, \texttt{w1}), (\texttt{a2}, \texttt{w2}), \ldots, (\texttt{an}, \texttt{wn})]$$

de parells `(ai,wi)` que contenen les lletres `ai` emeses per la font i els pesos `wi` (enter $\geqslant 1$) de cadascuna.

La funció ha de retornar el codi de Huffman binari canònic per a aquesta font.

Usant aquests codis de Huffman per a fonts creades a partir d'un text qualsevol es pot codificar el text, però per poder descodificar la seqüència binària que s'obté cal indicar al descodificador quin és el codi de Huffman usat, i per tant aquest codi s'ha d'incloure dins del fitxer comprimit, augmentant-ne la mida.

Això es pot evitar usant codis de Huffman adaptatius. Opcionalment podeu implementar aquest tipus de codificació amb funcions `huffman_encode(txt)` i `huffman_decode(cod)` que codifiquen un text i descodifiquen la cadena binària corresponent, recuperant el text original, de manera que el codi de Huffman es va creant i modificant (exactament de la mateixa manera) durant els processos de codificació i descodificació.

Codificació aritmètica. Escriviu una funció `arithmetic_encode(txt,k)` que faci la codificació aritmètica d'un string `txt` de la manera següent:

- primer crea una font a partir de les freqüències amb què apareixen les lletres a `txt`;
- a continuació el codifica amb codificació aritmètica entera treballant amb `k` bits.

Escriviu una funció `arithmetic_decode(code,k,src,len)` que sigui la inversa de l'anterior: a partir de la cadena binària codificada ha de recuperar el text original; aquí és necessari passar al descodificador la font usada per codificar ja que a partir de `code` no es pot saber quina és. Alternativament es podrien afegir els paràmetres `k` i `src` a la seqüència binària `code` i aleshores no caldria passar-los com a paràmetres però la seqüència augmentaria de mida. També se li ha de passar la longitud del text codificat.

Opcionalment podeu implementar la codificació/descodificació aritmètica adaptatives, i aleshores no és necessari crear la font ni per tant passar-la al descodificador.