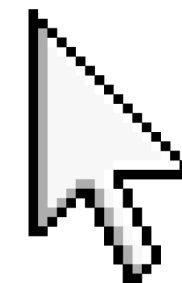


Project

HUFFMAN

DYNAMIC ALGORITHM

Start





THE CORE



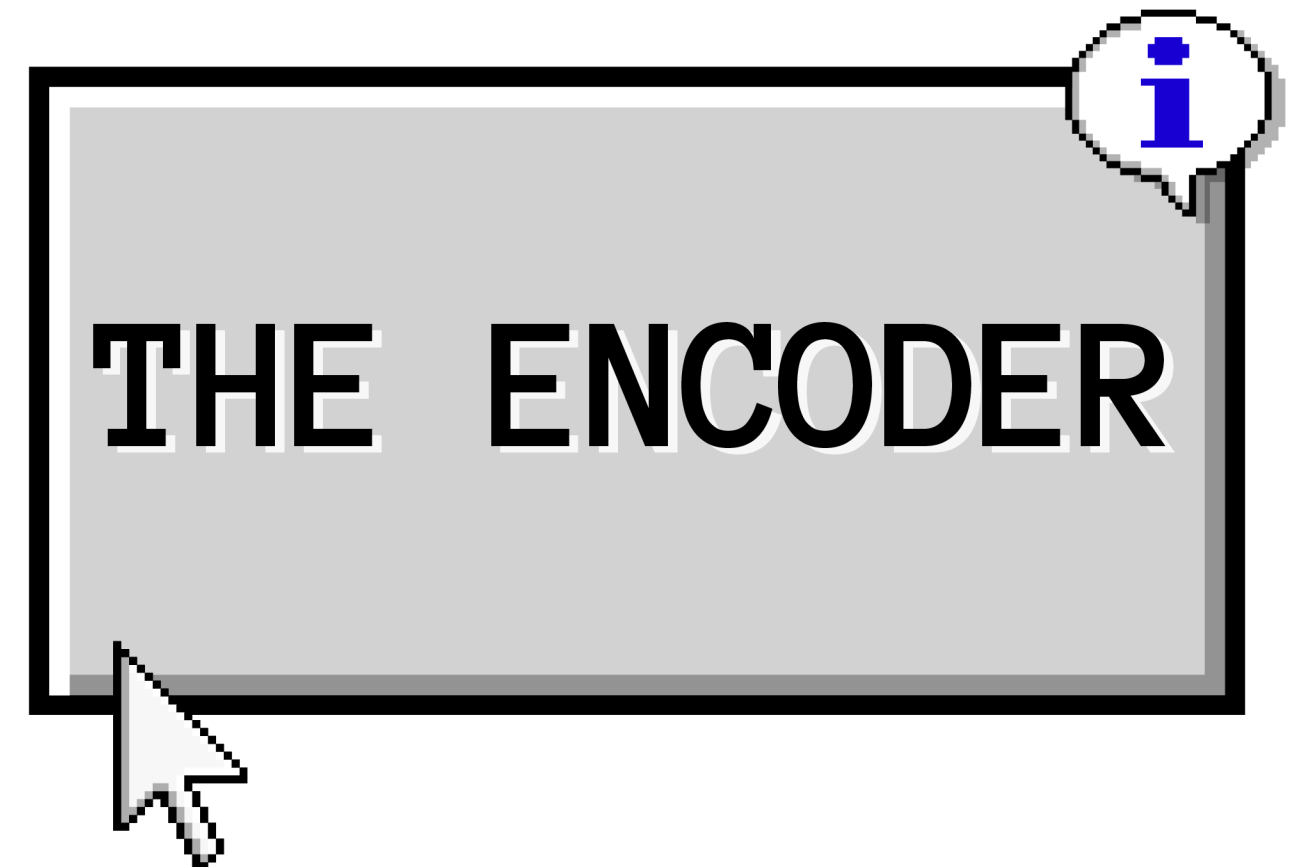
Architecture hybride:

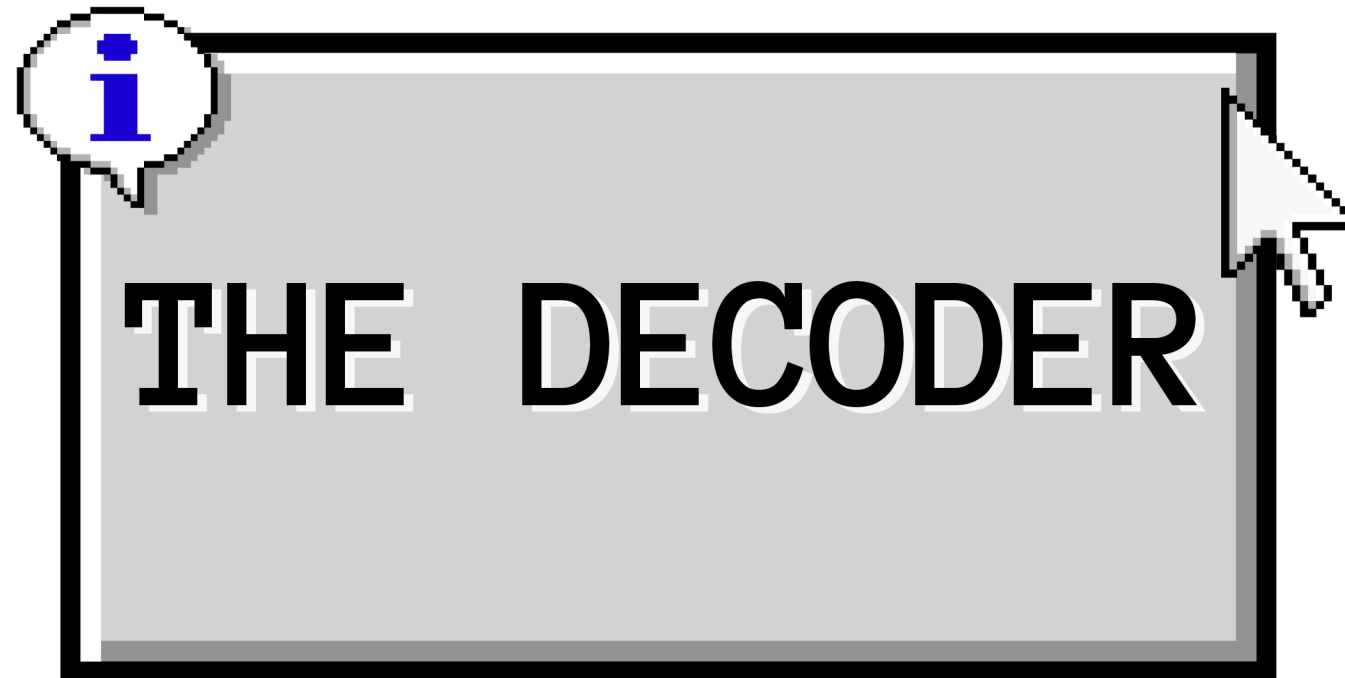
- arbre (nodes) :
parcours top-down
- hash-map (dictionnaire symbol_node) :
accès aux feuilles (encodage et mise
à jour de l'arbre)

Distinction explicite : LeafNode
(données) vs InternalNode (structure)

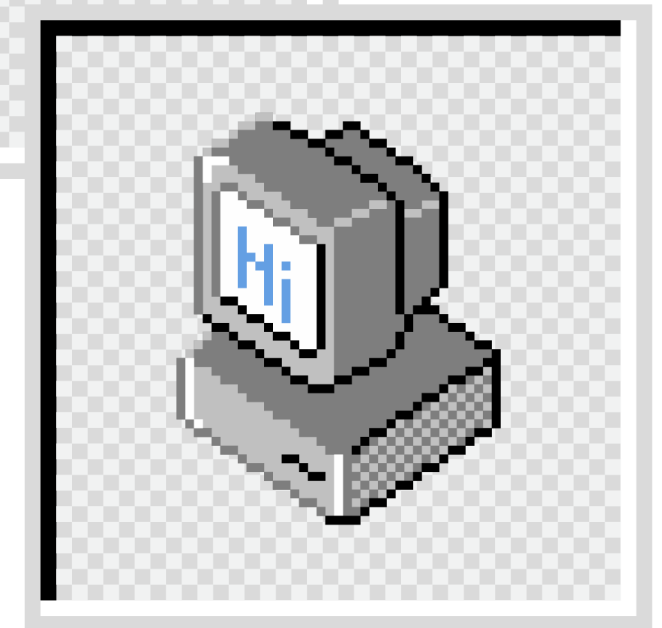
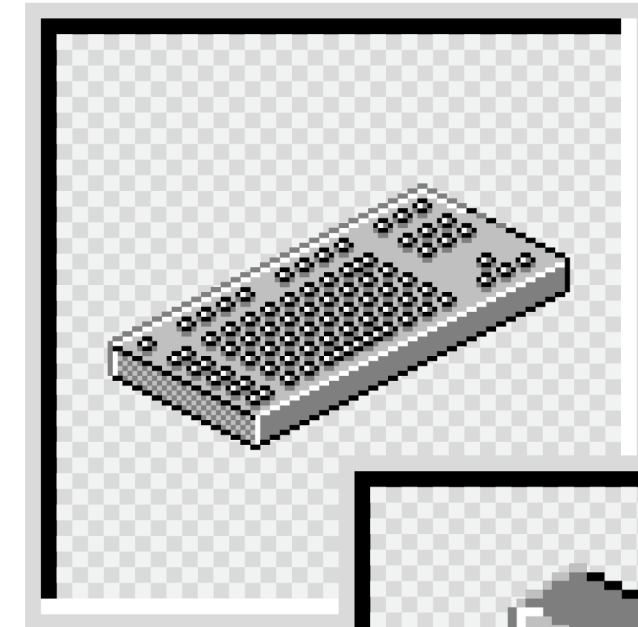


- Lecture binaire (rb) : support des caractères multi-octets (emojis, symboles...) et fichiers binaires
- Structure du fichier retourné :
[En-tête 64 bits] + [Flux Huffman]
+ [Padding]





- Même logique que l'encoder : symétrie
- Stratégie de lecture : en-tête puis boucle basée sur le comptage d'octets
- Ecriture du résultat dans un fichier en binaire (wb)





RESULTS

	file	original_size	compressed_size	ratio	compression_ms	decompression_ms
1	gutenberg-small-fr.txt	290	210	0.724	122.053	50.078
2	json-like-config.txt	170	143	0.841	19.670	19.415
3	short.txt	3	12	4.000	0.597	0.320
4	data-json-sample.txt	309	215	0.696	52.093	46.639
5	code-python-sample.txt	600	407	0.678	89.108	80.924
6	random-uniform-letters.txt	655	430	0.656	98.079	81.151
7	english-mixed-case.txt	202	173	0.856	35.935	38.842
8	emojis.txt	1758	1321	0.751	610.631	582.637
9	Blaise-Pascal.txt	118026	68594	0.581	38279.061	45327.403
10	random-biased-letters.txt	407	83	0.204	4.605	6.366

