1501 Final Exam

Huffman Compression

* Works on strings, characters and also image files.
* To fit R characters : need log R bits of storage PER BLOCK
* Code blocks are fixed in size and code words are variable
* To create this, make nodes of all the characters and their frequency. Add the shortest frequency together until you reach the root.
* K is the number of unique chars—the uncompressed size is the ceiling of log K times N
* We know for sure that its impossible to find a compression algorithm for each bitstream

LZW Compression and Expansion

* Lemple Ziv Welch – 3 old white men
* Variable length input and fixed length output
* While !EOF, match longest prefix in codebook, output codeword, append prefix and next character.
* Expansion is like opposite of the compression
* Read the codeword, find the pattern and output it and then use the previous pattern and the first character of the current pattern for next iteration.
* Only works because SAME codebook
* Better to use a trie than a hashtable
* Fewer bits 🡪 better compression, harder expansion
* Larger bits 🡪 not great compression, easy expansion

Entropy

* Measures the unpredictability of something
* How much of a surprise can a compressed file be
* Entropy of a character = -log of frequency
* 50% = - log ½ = -log 2^-1 = 1 bit
* 25% = - log ¼ = 2 bits

Integer Multiplication