CMSC262 DS/Alg Applied Blaheta

Homework 1

Due: 16 Sep 2019

Problem 1.1

Consider an image with eight colours, with the following counts for each colour:

transparent	12,000
opaque white	15,000
semitransparent white	6,000
light green	12,600
dark green	8,400
cyan	10,000
yellow	16,000
black	20,000

a. Build a Huffman tree for these frequency counts. Draw it.

b. The first non-metadata line of the file contains, in sequence, 2 transparent pixels, one semitransparent white pixel, two opaque white pixels, and five black pixels. Encode this sequence into the Huffman code implied by the tree you drew. Though this will technically just be ones and zeroes, indicate which groups of bits correspond to each pixel.

Problem 1.2

We run the Huffman algorithm on a set of symbols paired with their relative frequencies, to produce a trie that creates a correspondence between the symbols and variable-length binary strings. Consider now what would be required to actually implement the Huffman algorithm: identify (in the language of your choice or pseudocode) what class(es) or struct(s) you would need to create to represent the tree and support the algorithm (including their fields); and what library ADTs would you make use of, and how.

(Note: this is essentially the early design phase of a Huffman implementation, but that's all—no code is expected here! We talked about parts of this in class; fill out that discussion a bit.)

Collaboration policy: group work! If you work with other people on this homework, hand in one copy and put all your names on top. There will be a revision cycle for this.