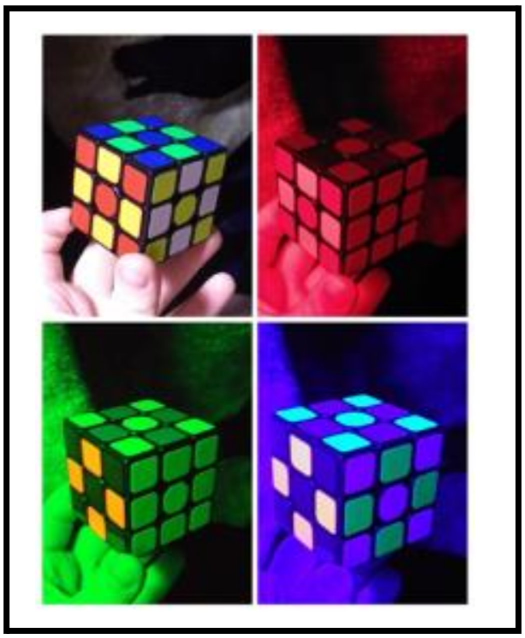
Steven Karl
CSCI576 - Multimedia Systems Design
Parag Halvadaar
10/10/2022

Assignment 2

Question1: Color Theory – 10 points

A Rubik's cube is a cube-shaped puzzle with 6 different 3x3 colored tiled sides: white, green, red, blue, orange, and yellow. The goal of the puzzle is to rotate sides and make each face have 3x3 tiles of the same color. When held under different colored lights (white, red, green, blue) the cube looks very interesting and vivid, see below:



Explain why this happened. Why do some tiles look bright, almost glowing, while others appear muted and devoid of their original color? *(4 points)*

Under white light each color will absorb all visible frequencies and reflect their respective color which is why each color will look as it typically does. However a color like blue or green under a red light will absorb all of the red light and as a result will look black/muted or devoid of color. In contrast a red colored tile under red light will reflect all of the red and appear to be a deeper red. A color like yellow under a green light will look brighter because it reflects all of the yellow frequencies of light that are contained by the green light being shined upon it. To summarize an object will absorb all light frequencies that it doesn't contain and reflect those that it does. This will cause it to glow or appear muted depending on the colors that you accentuate.

Assuming ideal conditions, you have the following lighting conditions to solve the puzzle – under pure yellow light or under red light. Which of these two light choices make it harder to solve? Give reasons for your choice of answer. *(6 points)*

Under both pure yellow light and red light the rubik's cube will be difficult to solve because red and yellow are both primary colors and have a large impact on the colors present in the rubiks cube.

Question 2: Color Theory (*10 points*)

The chromaticity diagram in (x, y) represents the normalized color matching functions X, Y and Z. Prove that: *(2 points)*

$$Z = [(1 - x - y)/y]Y$$

Solution:

$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

$$z = \frac{Z}{X+Y+Z}$$

Put Together:

$$X = x(X + Y + Z) \quad (1)$$

$$Y = y(X + Y + Z) \quad (2)$$

$$Z = z(X + Y + Z) \quad (3)$$

$$x + y + z = 1$$

$$\frac{y}{y} = 1 \ (Equation\ is\ key\ to\ breaking\ this\ down)$$

$$Z = z * (\frac{1}{y}) * y * (X + Y + Z)$$

$$Z = z * (\frac{1}{y}) * \frac{Y}{(X+Y+Z)} * (X + Y + Z)$$

$$Z = z * (\frac{1}{y}) * Y$$

$$Z = 1 - x - y * (\frac{1}{y}) * Y$$

$$Z = Y[(\frac{1-x-y}{y})]$$

Here you are tasked with mapping the gamut of a printer to that of a color CRT monitor. Assume that gamuts are not the same, that is, there are colors in the printer's gamut that do not appear in the monitor's gamut and vice versa. So in order to print a color seen on the monitor you choose the nearest color in the gamut of the printer. Answer the following questions

1. Comment (giving reasons) whether this algorithm will work effectively? *(2 points)*

Yes, I believe that it will still work effectively. While the gamuts don't match exactly they are still rooted in the typical X, Y, Z primaries. One of the gamuts may be wider or narrower than another however the general color will be preserved with slight changes in hue, saturation, and value.

The X, Y, Z primaries are not tied to a physical display device and they sit outside of the visible gamut which means they completely enclose the visible gamut. This means that X, Y, Z primaries can be used to define colors independent of display technology. Since both gamuts are enclosed within the area defined by the X, Y, Z primaries they will share similarities that will allow for a general but not necessarily exact translation between the two gamuts.

2. You have two images – a cartoon image with constant color tones and a real image with varying color tones? Which image will this algorithm perform better – give reasons? *(2 points)*

I believe it will perform better for the cartoon image with constant color tones.

While the final image might be off in its color selection(Slight difference in hue saturation and value) it will spread throughout the whole image creating an image that looks similar but is slightly off in terms of hue, saturation and value. By this I mean that if the cartoon image was one shade of red and we selected the nearest color as a different shade of red it would still resemble the desired image. Also if the nearest color wasn't even a red but let's say a dark orange you would still be able to tell what the resulting image is.

However, for varying color tones, while there is a chance that one of the tones might match more closely between the monitor's gamut and the printer's gamut, there is also more of a chance that you won't find reasonable matches. You could end up with a very textiled/pixelated image that might not even resemble the image on the monitor.

3. Can you suggest improvements rather than just choosing the nearest color? *(4 points)*

Yes, averaging the surrounding colors to get something that is a closer match hence if its lighter average the colors nearby that tend to look lighter. Also interpolating might make sense because if you know two colors from the monitor aren't the same but you get the same nearest color instead you could linearly interpolate to get a slightly different lighter or darker shade because you know that it shouldn't be exactly that color.

Question 3: Entropy Coding *10 points*

Consider a communication system that gives out only two symbols X and Y. Assume that the parameterization followed by the probabilities are:

$$P(X) = x^k \; and \; P(Y) = (1 - x^k)$$

1. Write down the entropy function and plot it as a function of *x* for k=2. (*1 points)*

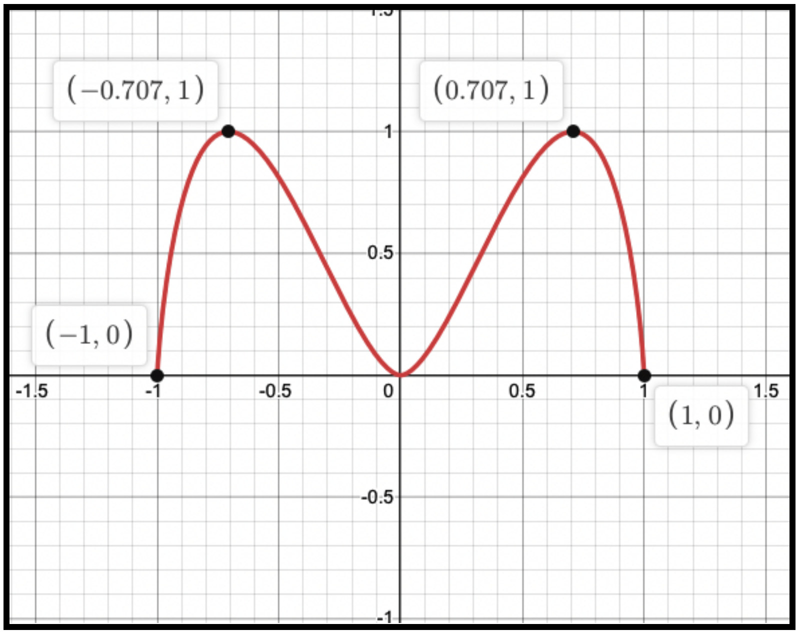
Solution:

Given:

$$H(x) = -\sum_{1}^{n} P(i)log_2P(i)$$

$$H = -x^2log_2x^2 - (1 - x^2)log_2(1 - x^2)$$

Plot:



2. From your plot, for what value of *x* with *k*=2 does H become a minimum? (*1 points)*

From the plot shown above it can be seen that H becomes a minimum at x = 0 or x = +/- 1.

3. Your plot visually gives you the minimum value of *x* for *k*=2, find out a generalized formula for x in terms of *k* for which H is a minimum *(3 points)*.

   Solution:

From class "Entropy is small (always ≥ 0) when some symbols that are much more likely to appear than other symbols".

Therefore, Entropy H(x) is at a minimum when:

$P(X) = 1;\ Case\ 1$

$P(Y) = 1;\ Case\ 2$

Case 1:

$$P(X) = x^2 = 1$$

$$x = \sqrt{1} = \pm 1$$

Case 2:

$$P(Y) = 1 - x^2 = 1$$

$$x^2 = 0$$

$$x = \sqrt{0} = 0$$

Therefore, entropy will be at a minimum when x is equal to +/- 1 or 0.

4. From your plot, for what value of *x* with *k*=2 does H become a maximum? (*1 points)*

From the plot shown above it can be seen that H becomes a maximum when x = +/- 0.707.

5. Your plot visually gives you the maximum value of *x* for *k*=2, find out a generalized formula for x in terms of k for which H is a maximum *(4 points)*.

Solution:

Based on the logic from the lecture: “H is highest (equal to log 2 N) if all symbols are equally probable”. Given that we can set the P(X) and P(Y) equal to solve for the x that satisfies this.

$$P(X) = P(Y)$$

$$x^2 = (1 - x^2)$$

$$2x^2 = 1$$

$$x^2 = \frac{1}{2}$$

$$x = \sqrt{\frac{1}{2}}$$

$$x = \pm 0.707$$

Question 4: Huffman Coding/Entropy *10 points*

Bob has a pen pal, Alice, who has been learning about information theory and compression techniques. Alice decides from now on that they should exchange letters as encoded signals so they can save on ink. The following is a letter that Alice sends to Bob on her trip to Paris:

> Dear Bob,
> *Hello from Paris!*
> *I got this postcard from the Louvre. You would love Paris! I hope to hear from you.*
> Alice

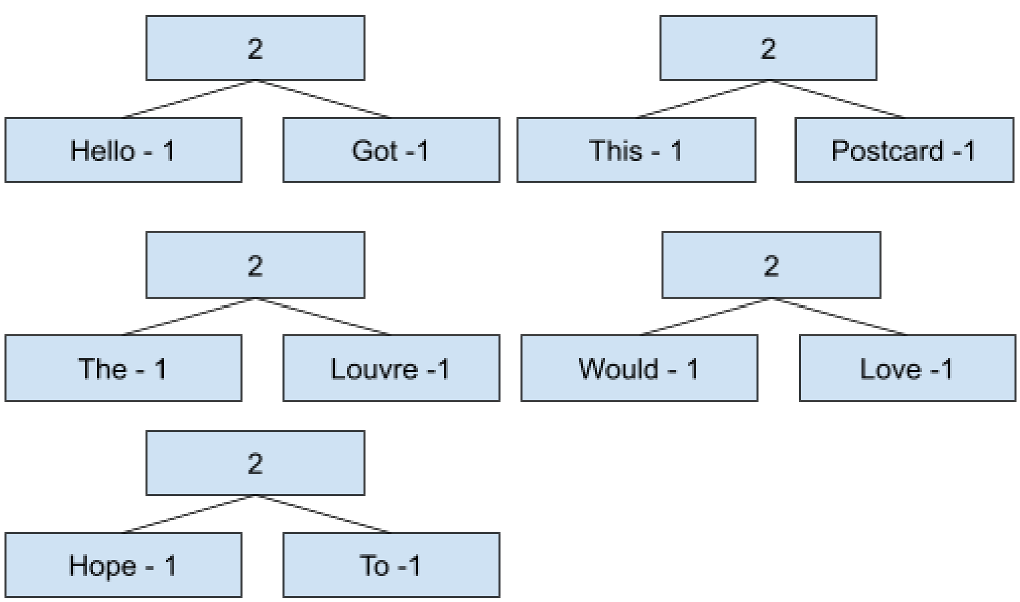
Find and show a Huffman code for the body of Alice's postcard (i.e. exclude "Dear Bob" and "Alice"). Treat each word as a symbol, and don't include punctuation. What is the average code length? (*3 points*)

Solution:

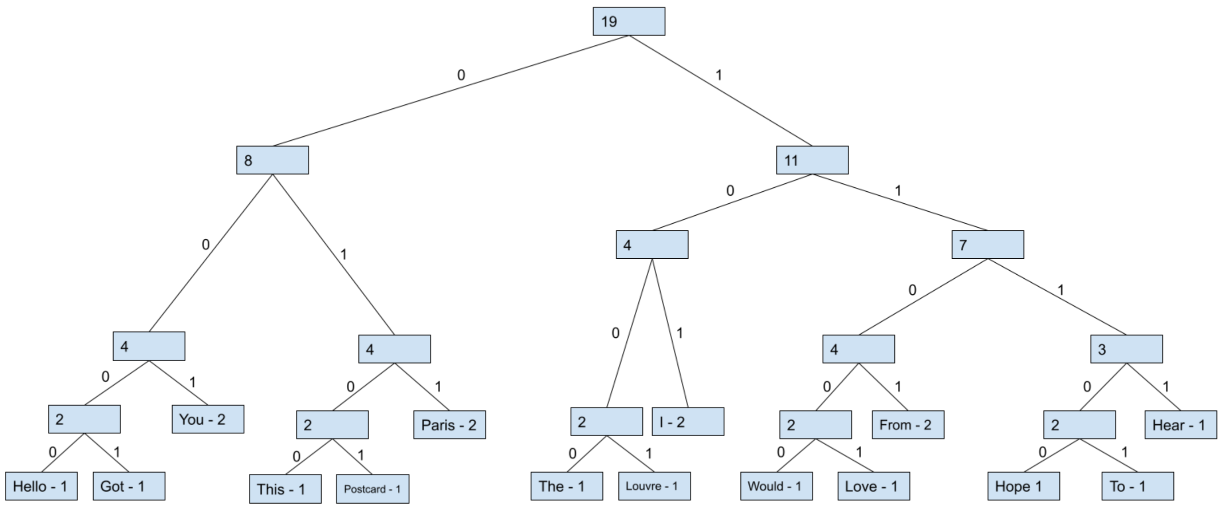
Step 1: Symbol - Frequency - Probability

- Hello - 1
- From - 2
- Paris - 2
- I - 2
- Got - 1
- This - 1
- Postcard - 1
- The - 1
- Louvre - 1
- You - 2
- Would - 1
- Love - 1
- Hope - 1
- To - 1
- Hear - 1

Step 2: Combine min frequencies to create internal tree components



Now use minimums to create the rest of the tree shown below:



Step 4: Use tree to get codes

Symbol - Code (Length)

- Hello - 0000 (4)
- From - 1101 (4)
- Paris - 011 (3)
- I - 101(3)
- Got - 0001 (4)
- This - 0100 (4)
- Postcard - 0101 (4)
- The - 1000 (4)
- Louvre - 1001 (4)
- You - 001 (3)
- Would - 11000 (5)
- Love - 11001 (5)
- Hope - 11100 (5)
- To - 11101 (5)
- Hear - 1111 (4)

Step 5: Final Message and Calculate Average Code length

Final Message: I used the same return lines for ease of reading

0000 1101 011

101 0001 0100 0101 1101 1000 1001 001

11000 11001 011 101 11100 11101 1111 1101 001

$$Average\ Code\ Length = Sum/Count = \frac{5(4) + 4(10) + 3(6)}{20} = \frac{78}{20} = 3.9$$

Bob, having just learned about the [telegram](#) in history class, suggests to Alice that they can try writing their letters [as telegram messages](#) to shorten them even more. He sends Alice what her postcard might look like as a telegram:

> IN PARIS POSTCARD FROM LOUVRE *STOP*
> YOU WOULD LOVE *STOP*
> HOPE HEAR FROM YOU *STOP*

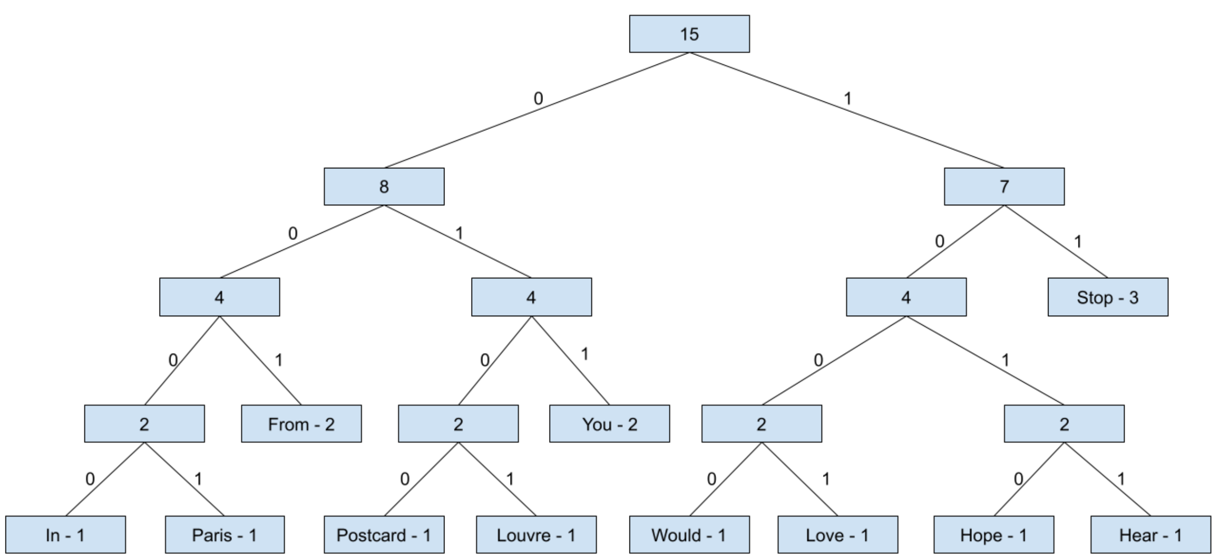
1. Find a Huffman code for the telegram message. What is the average code length? How does it compare to the original letter? (*3 points)*

   Solution:

   Step 1: Symbol - Frequency

   - In - 1
   - Paris - 1
   - Postcard - 1
   - From - 2
   - Louvre - 1
   - Stop - 3
   - You - 2
   - Would - 1
   - Love - 1
   - Hope - 1
   - Hear - 1

Step 2: Combine to create tree



Step 3: Symbol - Code (Count)

- In - 0000 (4)
- Paris - 0001 (4)
- Postcard - 0100 (4)
- From - 001 (3)
- Louvre - 0101 (4)
- Stop - 11 (2)
- You - 011 (3)
- Would - 1000 (4)
- Love - 1001 (4)
- Hope - 1010 (4)
- Hear - 1011 (4)

Step 4: Final Message and Calculated Code Length

Final Message: Used the same return locations for ease of reading

0000 0001 0100 001 0101 11

011 1000 1001 11

1010 1011 001 011 11

$$Average\ Length = \frac{Sum\ of\ bits\ used}{Total\ word\ count} = \frac{4(8) + 3(4) + 2(3)}{15} = \frac{50}{15} = 3.333$$

2. Which version of the message, postcard, or telegram, contains more information? Show quantitatively and explain qualitatively where the difference (if any) comes from. (*4 points)*

The postcard contains more information. This can be seen by looking at the average code length which in the case of the postcard is 3.9 while the average code length for the telegram is 3.33. When you look at the total bits in each message you can see that it's 78 bits for the postcard and 50 bits for the telegram. Additionally the telegram contains less unique words then the postcard 11 vs 15 respectively. Also the overall postcard message contains more total words/information then the telegram message.