

365 Programming assignment 2 report

Alen Mehmedbegovic

March, 8th 2024

1 Question 1

After making our python script, this is what is printed out.

```
[101, -11, -1]
```

2 Question 2

Implemented the half-tone dithering covered in class.

```
Half tone dither:
[[1 0 1 0 1 0 1 0]
 [0 0 0 0 0 0 0 0]
 [1 0 1 0 1 0 1 0]
 [0 0 0 0 0 0 0 0]
 [1 0 1 0 1 0 1 0]
 [0 0 0 0 0 0 0 0]
 [1 0 1 0 1 0 1 0]
 [0 0 0 0 0 0 0 0]]

Ordered dither:
[[1 0 1 0]
 [0 0 0 0]
 [1 0 1 0]
 [0 0 0 0]]
```

3 Question 3

3.1 (1)

```
First order entropy: 1.0
Second order entropy: 2.0
```

3.2 (2)

```
Average single Codeword length: 8.0 bits per symbol
Average dual Codeword length: 16.0 bits per symbol
```

3.3 (3)

Because the probabilities of each symbol (for single and dual) all add up to 1, the average codeword length will always be the bits of the char's needed to represent the symbol. The codeword length calculation will sum all the probabilities multiplied by the bits needed for the amount of chars used. Another observation is that since the input string will always be length 2, our implementation that reads it from left to right will also be the same reading it from right to left, as because there are only even strings, there will never be a character cut off with an odd length string. So all possible dual symbols remain the same.

3.4 (4)

First input: 'ADEECABD'

```
First order entropy: 1.75
Second order entropy: 2.625
Average single Codeword length: 6.0 bits per symbol
Average dual Codeword length: 9.142857142857142 bits per symbol
```

Second input: 'BDCCCEEA'

```
First order entropy: 1.6556390622295665  
Second order entropy: 2.48345859334435  
Average single Codeword length: 6.0 bits per symbol  
Average dual Codeword length: 9.142857142857142 bits per symbol
```

Third input: 'AABCCDBCBDCCCEEA'

```
First order entropy: 1.811278124459133  
Second order entropy: 3.1697367178034836  
Average single Codeword length: 7.0 bits per symbol  
Average dual Codeword length: 12.799999999999999 bits per symbol
```

It seems like the larger the input is, the higher the entropy will be and a higher codeword length of bits/symbol.
It also seems like the length of the string dictates the codeword length, rather than the actual probabilities and distributions of characters. Test1 and test2 the have same results despite being different strings.