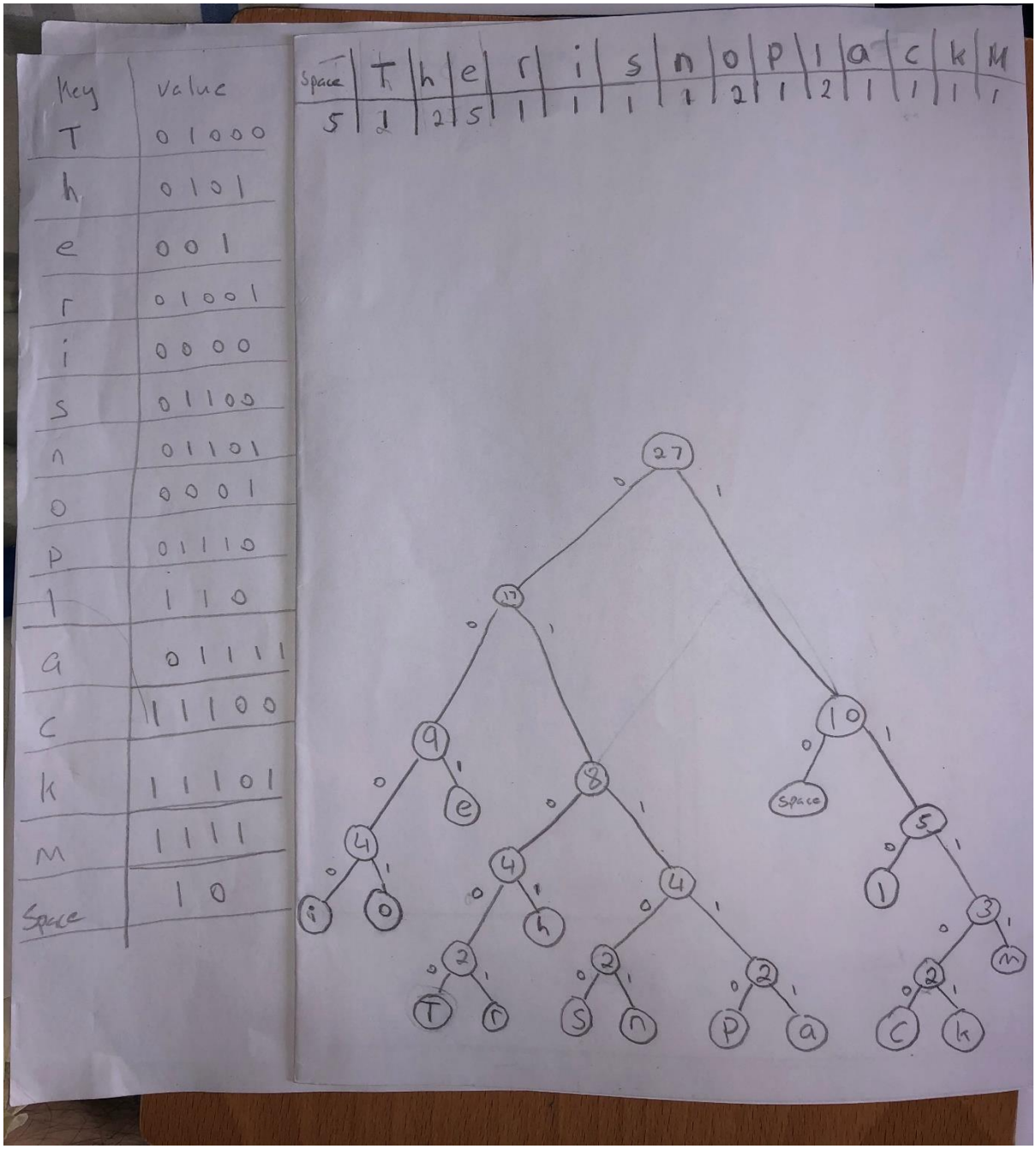
# COMP20290
# Algorithms

## Assignment 1

Morgan Collins McCarthy 18761429



First, I tested to see if my program was running correctly.

```
C:\Users\Morgan\Workspace\Huffman\src>java BinaryDump 240 < test.txt
01010100010001010101001101010100010101000100010101010011010101000101010001000101010100110101010001010100010001010101001101010100
128 bits
```

Here I get the original bits of the test file.

```
C:\Users\Morgan\Workspace\Huffman\src>java HuffmanAlgorithm c < test.txt | java BinaryDump 240
0101010100010100010110101010011000000000000000000000000000010000010110010110010110010110000
88 bits
```

I then compress the file and get 88 bits.

```
C:\Users\Morgan\Workspace\Huffman\src>java HuffmanAlgorithm c < test.txt | java HuffmanAlgorithm d | java BinaryDump 240
01010100010001010101001101010100010101000100010101010011010101000101010001000101010100110101010001010100010001010101001101010100
128 bits
```

Next, I compress the file and the decompress it and we can see I get the original 128 bits.

```
C:\Users\Morgan\Workspace\Huffman\src>java HuffmanAlgorithm c < test.txt | java HuffmanAlgorithm
TESTTESTTESTTEST
```

Finally, I compress the file and decompress it to see if I get the original text which is correct.

| File | Compression Time (nano) | Compressed Size | Original Size | Ratio | Decompress Time |
|---|---|---|---|---|---|
| genomeVirus.txt | 17684200 | 12576 | 50008 | 0.2514798 | 8795400 |
| medTale.txt | 23151500 | 23912 | 45056 | 0.5307173 | 13215600 |
| mobydick.txt | 3080514500 | 5341208 | 9531704 | 0.5603623 | 2264261100 |
| q32x48.bin | 5347500 | 816 | 1536 | 0.53125 | 5211900 |

I then when on to compress each of the files provided to a new compressed txt file and recorded the time in nano seconds it took to compress. I recorded the original file sizes and the compressed sizes. I then decompressed each file and recorded the time it took. I also made sure the decompressed bits matched the original bits.

```
C:\Users\Morgan\Workspace\Huffman\src>java HuffmanAlgorithm c < mobydickCompressed.txt > d.txt
```

Compression files again can result in the file getting larger or smaller , the file can be smaller as a result of an algorithm not being able to perform omniscient perfect compression. This depends on the algorithm and the file being compressed.

Using run length encoding on q32x48.bin we get a result of:

```
1144/1536 = 0.74479166666
```

Huffman we get a result of:

```
816/1536 =0.53125
```

We see Huffman compression give us a better compression ratio in this case. This is because RLE works better when there is a lot of repeated data. RLE would compress better than Huffman if there were more repeated bits in the bit map.