Question 1

Huffman coding works such that it assigns shorter codelengths to characters of higher frequencies vs less appearing characters.

In File A, there are M unique characters each appearing N/M times.

In File B, there are M unique characters with the ith character appearing 2^i times. Over a large N, for the same unique character “J”, the Huffman coding for file B would exist such that the “J” would be closer to the root than in the Huffman coding for File A. As this property increases for each unique m, in file B would require a smaller quantity of bits than file A. Additionally, for file A, the equal frequency , despite the efficiency of the Huffman coding, causes each of the unique characters to have approximately same length encodings. This overall increases the number of bits thus storage required for File B. The more frequent characters obtain short encodings whilst the less frequent ones obtain longer encodings. Overall **File B**, thus achieves a higher compression than File A. Assuming both files have the same initial size, File B would have a higher compression ratio since the compressed file B has a smaller size than the compressed File A.

Question 2

Data for oak-ridge.jpg

```
name: oak-ridge.jpg
compress from oak-ridge.jpg to UHFoak-ridge.jpg.hf
file: 4027336 bits to 4018768 bits
read 4027336 bits, wrote 4018768 bits
bits saved = 8568
Compress took 50 milliseconds%
Compression Ratio = 1.002131996671617
```

Data for mtblanc.jpg

```
name: mtblanc.jpg
compress from mtblanc.jpg to UHFmtblanc.jpg.hf
file: 2624056 bits to 2616920 bits
read 2624056 bits, wrote 2616916 bits
bits saved = 7140
Compress took 45 milliseconds%
Compression Ratio = 1.002726869755285
```

Comparing mtblanc.jpg and oak-ridge.jpg, they have similar compression ratios possible due to same file type. However, their compression times are directly proportional to the original size of the file. From the data, mtblanc takes a relatively shorter time due to its smaller size.

Data for hawthorne.txt

```
name: hawthorne.txt
compress from hawthorne.txt to UHFhawthorne.txt.hf
file: 3974152 bits to 2229704 bits
```

```
read 3974152 bits, wrote 2229700 bits
bits saved = 1744452
Compress took 51 milliseconds%
Compression Ratio = 1.782367525016774
```

Comparing all three empirical evidences, the text based file has a higher compression ratio and this aligns with the expectation that text is likelier to contain predictable patterns and redundancies. Additionally, the compression of text is less complex compared to the image based files. Compression time analysis is consistent with the claim that the compression time is dependent on the file size . The compression time for hawthorne is significantly higher than the image files, further indicating that text files are highly compressible wheres there is almost minimal compression for the image files. Text file compression might take slightly longer than image compression due to their larger size and complexity. However, in this case, the difference in compression time is not significant.

Question 3

Data for oak-ridge.jpg

```
uncompress from UHFoak-ridge.jpg.hf to DUHFoak-ridge.jpg.uhf
file: 4018768 bits to 4027336 bits
read 4018768 bits, wrote 4027336 bits
8568 compared to 8568
Decompress took 48 milliseconds%
```

Data for mtblanc.jpg

```
uncompress from UHFmtblanc.jpg.hf to DUHFmtblanc.jpg.uhf
file: 2616920 bits to 2624056 bits
read 2616920 bits, wrote 2624056 bits
7136 compared to 7136
Decompress took 33 milliseconds%
```

Data for hawthorne.txt

```
uncompress from UHFhawthorne.txt.hf to DUHFhawthorne.txt.uhf
file: 2229704 bits to 3974152 bits
read 2229704 bits, wrote 3974152 bits
1744448 compared to 1744448
Decompress took 33 milliseconds%
```

The overall decompression times for all three files are shorter than their initially obtained compression times; thus the decompression runtime for all three files are relatively fast. There seems to be a codependence on file size however this is not conclusive since for different file types of different sizes (hawthorne and mtblanc), decompression takes the same amount of time. The file with the largest decompression runtime is oak-ridge and it is also the file of the largest size. This may show that decompression runtime may depend on size. For hawthorne and mtblanc, even though hawthorne is larger than mtblanc, they have the same decompression runtime further bolstering the fact that text is more compressible. Additionally, it seems some bit was accounted for, where 7136 bit was recovered during decompression from an earlier saved 7140 during compression.