Assignment 2
Algorithms

Name	ID
البدري محمد السيد البدري حسن عبدالله	20010329

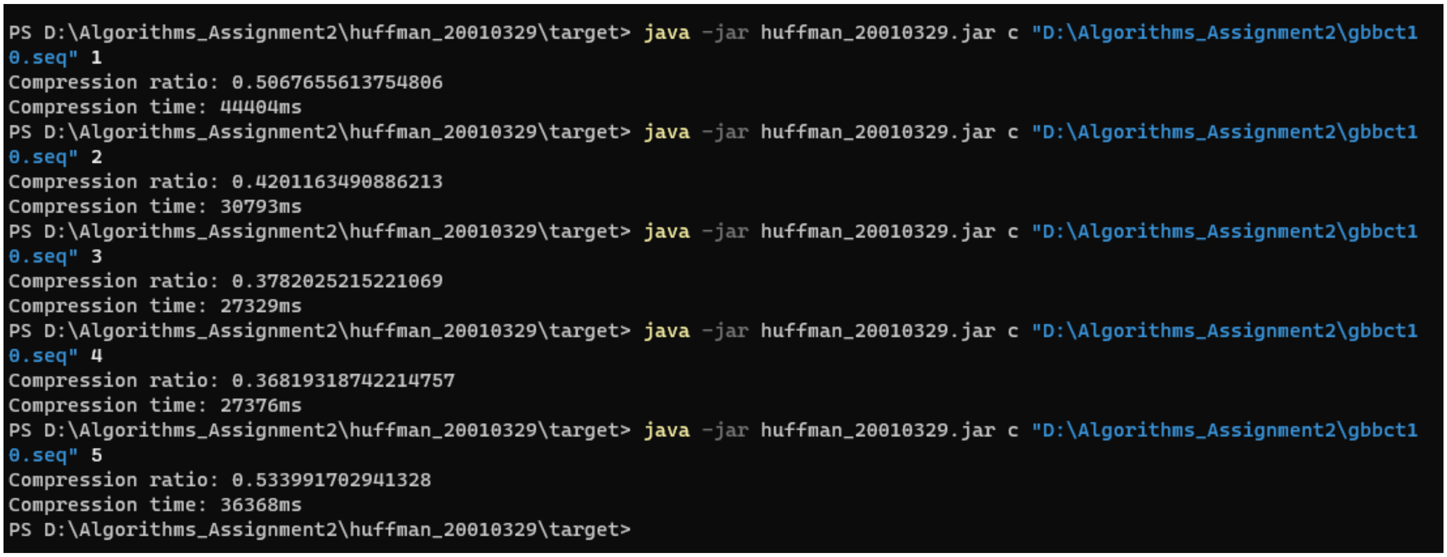
“I acknowledge that I am aware of the academic integrity guidelines of this course, and that I worked on this assignment independently without any unauthorized help”.

1. Analysis of Huffman Algorithm

- File 1: gbbct10.seq (size = 471 MB)

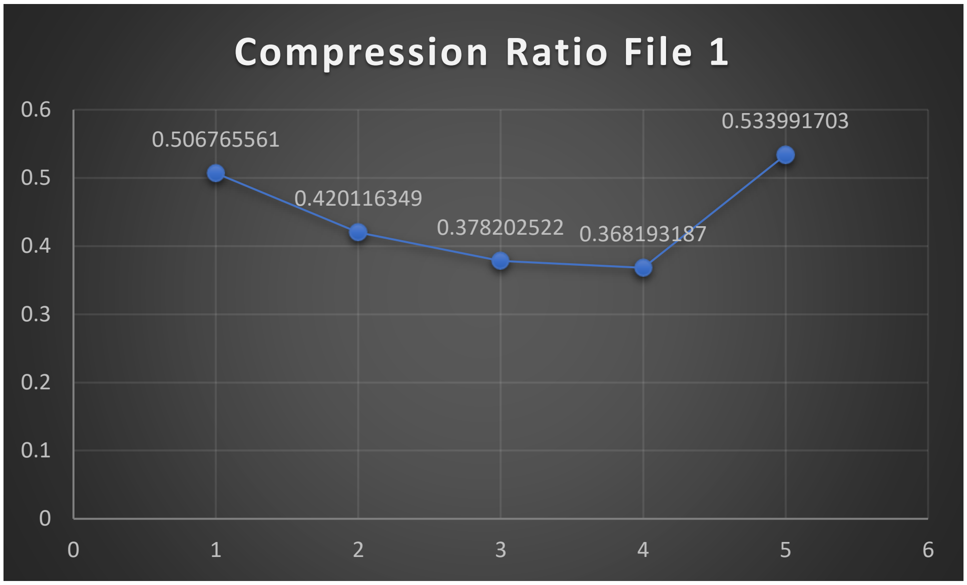
Chunk Size “n”	Compression Ratio
1	0.5067655613754806
2	0.4201163490886213
3	0.3782025215221069
4	0.36819318742214757
5	0.533991702941328

- 7 z Compression Ratio = 112 MB / 471 MB = 0.23



```
PS D:\Algorithms_Assignment2\huffman_20010329\target> java -jar huffman_20010329.jar c "D:\Algorithms_Assignment2\gbbct1
0.seq" 1
Compression ratio: 0.5067655613754806
Compression time: 44404ms
PS D:\Algorithms_Assignment2\huffman_20010329\target> java -jar huffman_20010329.jar c "D:\Algorithms_Assignment2\gbbct1
0.seq" 2
Compression ratio: 0.4201163490886213
Compression time: 30793ms
PS D:\Algorithms_Assignment2\huffman_20010329\target> java -jar huffman_20010329.jar c "D:\Algorithms_Assignment2\gbbct1
0.seq" 3
Compression ratio: 0.3782025215221069
Compression time: 27329ms
PS D:\Algorithms_Assignment2\huffman_20010329\target> java -jar huffman_20010329.jar c "D:\Algorithms_Assignment2\gbbct1
0.seq" 4
Compression ratio: 0.36819318742214757
Compression time: 27376ms
PS D:\Algorithms_Assignment2\huffman_20010329\target> java -jar huffman_20010329.jar c "D:\Algorithms_Assignment2\gbbct1
0.seq" 5
Compression ratio: 0.533991702941328
Compression time: 36368ms
PS D:\Algorithms_Assignment2\huffman_20010329\target>
```

Observation:



- Compression Ratio value decreased till n = 4, then it started to increase at n = 5
- Compression Ratio for 7z was better than compression Ratio using Huffman for different values of n.
- Note that: Compression Ratio = compressed size / original size, which means that small compression ratio is better than larger compression ratio.

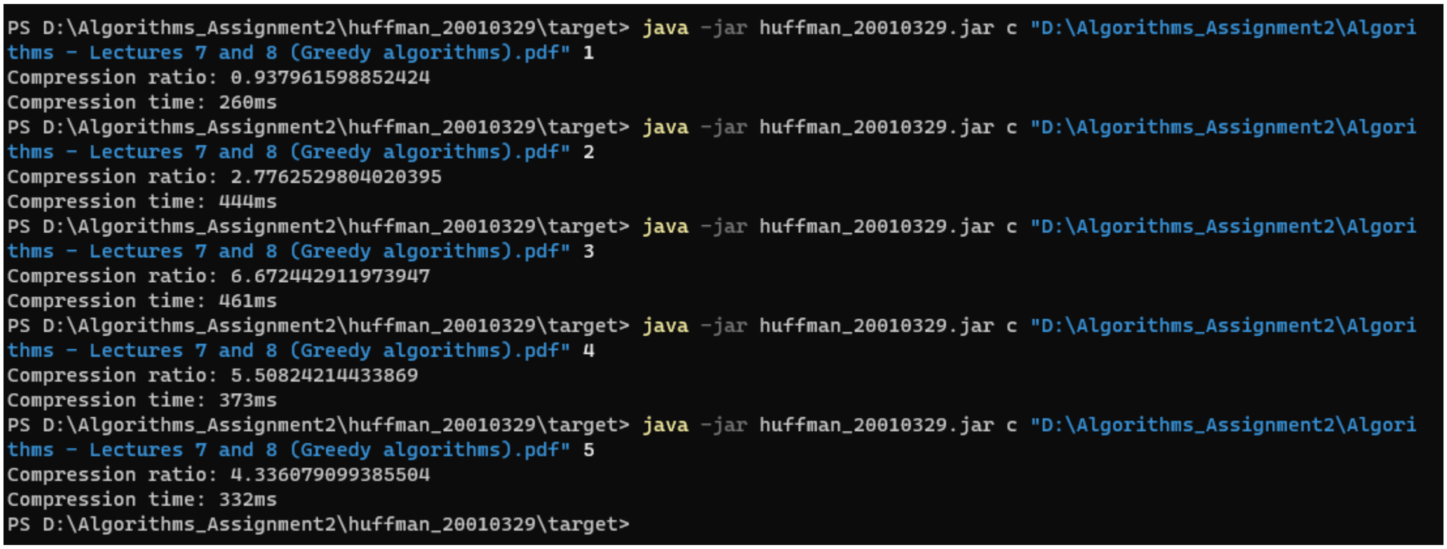
Conclusion:

- It goes like upside-down bell curve, by increasing chunk size n, the size of the compressed file decreases till certain point then the size compressed file size starts to increase due to increases the size of the header due to storing a lot of combinations for the chars stored in that file with their codes.
- Compression Ratio for 7z was better than compression Ratio using Huffman. This is because how different compressing algorithms deal with the files, Huffman depends on the utilization of the frequency of chunks in the file, while other algorithms may depend on different parameters to compress the file perfectly.

- File 2: Algorithms - Lectures 7 and 8 (Greedy algorithms).pdf (Size = 806 KB)

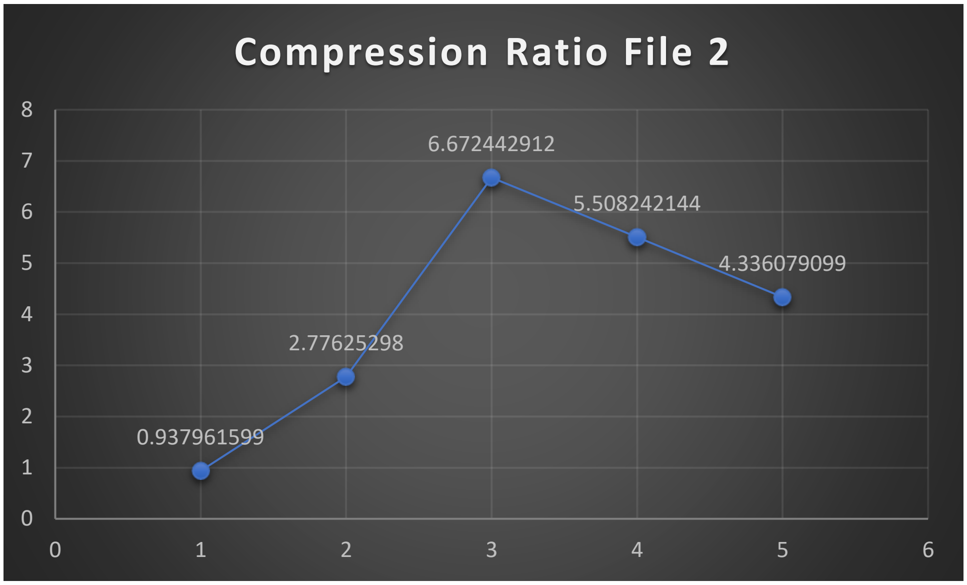
Chunk Size “n”	Compression Ratio
1	0.937961598852424
2	2.7762529804020395
3	6.672442911973947
4	5.50824214433869
5	4.336079099385504

- 7Z Compression Ratio = 553 KB / 806 KB = 0.686



```
PS D:\Algorithms_Assignment2\huffman_20010329\target> java -jar huffman_20010329.jar c "D:\Algorithms_Assignment2\Algori
thms - Lectures 7 and 8 (Greedy algorithms).pdf" 1
Compression ratio: 0.937961598852424
Compression time: 260ms
PS D:\Algorithms_Assignment2\huffman_20010329\target> java -jar huffman_20010329.jar c "D:\Algorithms_Assignment2\Algori
thms - Lectures 7 and 8 (Greedy algorithms).pdf" 2
Compression ratio: 2.7762529804020395
Compression time: 444ms
PS D:\Algorithms_Assignment2\huffman_20010329\target> java -jar huffman_20010329.jar c "D:\Algorithms_Assignment2\Algori
thms - Lectures 7 and 8 (Greedy algorithms).pdf" 3
Compression ratio: 6.672442911973947
Compression time: 461ms
PS D:\Algorithms_Assignment2\huffman_20010329\target> java -jar huffman_20010329.jar c "D:\Algorithms_Assignment2\Algori
thms - Lectures 7 and 8 (Greedy algorithms).pdf" 4
Compression ratio: 5.50824214433869
Compression time: 373ms
PS D:\Algorithms_Assignment2\huffman_20010329\target> java -jar huffman_20010329.jar c "D:\Algorithms_Assignment2\Algori
thms - Lectures 7 and 8 (Greedy algorithms).pdf" 5
Compression ratio: 4.336079099385504
Compression time: 332ms
PS D:\Algorithms_Assignment2\huffman_20010329\target>
```

Observation:



- The compression ratio increased till n = 3, then it started to decrease again.
- Compression Ratio for 7z was better than compression Ratio using Huffman for different values of n.

Conclusion:

- Since the file size is relatively small then by increasing n leads to an increase in the size of the header of the file which is relatively larger than the size of the compressed file itself without header.
- The compression ratio started to decrease again which means that till a certain threshold the data in the header file would be little bit useful to compress the file.
- But the best chunk size was n = 1.
- Compression Ratio for 7z was better than compression Ratio using Huffman. This is because how different compressing algorithms deal with the files, Huffman depends on the utilization of the frequency of chunks in the file, while other algorithms may depend on different parameters to compress the file perfectly.