**Student Name: Mohammad Ali Agharazi Dormani**

**Student Number:19205451**

**SMS Encoding Rules: the rules are written bellow the table for each row**

**Question 1**

**To what extent did you rely on a common understanding of SMS shorthand? Answer:**

to some extent because most people can understand it, I also wrote some words with fewer letters for instance home and hom are read the same way so there is no point in using that extra letter. More on this, is explained under the

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Original Message** | **Original Size** | **Compressed Message** | **Compressed Message Size** | **Compression %** | **Good** | **Fair** | **Poor** |
| I would like to see you tomorrow, lets meet at starbucks at 12 | 50 | Meet u 2morow @ 12, Sbux | 18 | (50-18)/50 =0.64 =64% | 4 | 0 | 1 |
| I am going to see my friend Hannah today; I will be a bit late. | 49 | I C Hannah 2day,im L8 | 17 | (49-17)/49= 0.653 =65.3% | 5 | 0 | 0 |
| Can you help me with my Programming and Matrix Algebra homework? | 54 | Can u hlp me w/ my coding & Mtrx HW? | 27 | (54-27) /54= 0.5=50% | 3 | 2 | 0 |
| Could you please, grab a bottle of red Shiraz wine on your way home? | 55 | Can u get a red Shraz on ur way hom? | 26 | (55-26) /55= 0.527=52.7% | 4 | 0 | 1 |
| I am going out to the movie theater with my friends | 41 | Im going 2 movie w/ frinds | 21 | (41-21)/41=0.488  =48.8% | 1 | 2 | 2 |

**RULES for row #1**

“I would like to see you” => “Meet u”

“tomorrow”=> “2morow”

“at” => “@”

“starbucks” => “Sbux”

**RULES for row #2**

“see” => “C”

“today”=> “2day”

“I will be a bit late” => “im L8”

**RULES for row #3**

“you” => “u”

“help” => “hlp”

“with” => “w/”

“programming” => “coding”

“and” => “&”

“Matrix algebra” => “Mtrx”

“homework” => “HW”

**RULES for row #4**

“could” => “can”

“you” => “u”

“please” => “”

“grab” => “get”

“Shiraz” => “Shraz”

“your” => “ur”

“home” => “hom”

**RULES for row #5**

“I am” => “Im”

“going out” => “going”

“to” => “2”

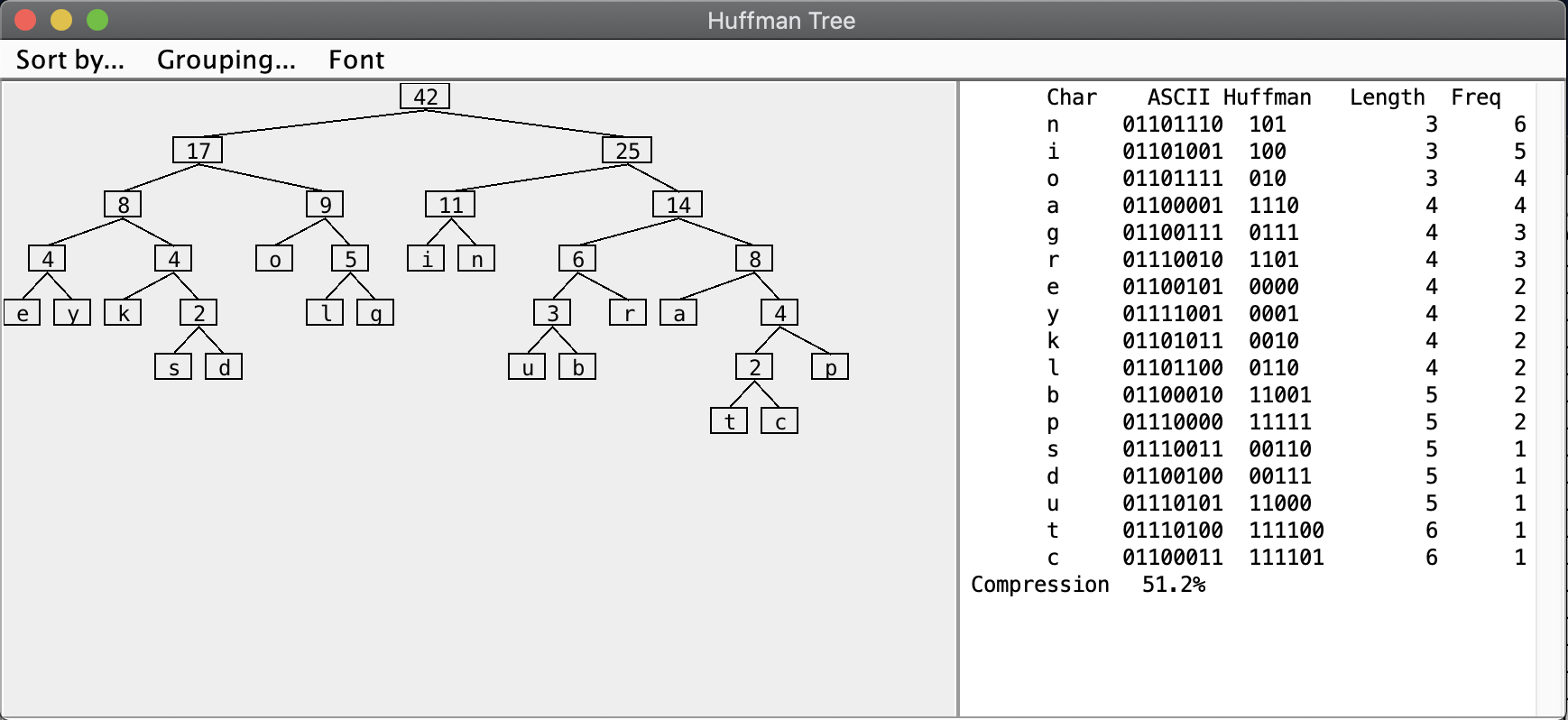
“movie theater” => “movie”

“with” => “w/”

“my friends” => “frinds”

|  |  |  |
| --- | --- | --- |
| **Letter** | **Code** | **Equivalent Code** |
| **a** | **1110** | **01100001** |
| **b** | **11001** | **01100010** |
| **c** | **111101** | **01100011** |
| **d** | **00111** | **01100100** |
| **e** | **0000** | **01100101** |
| **f** |  |  |
| **g** | **0111** | **01100111** |
| **h** |  |  |
| **i** | **100** | **01101001** |
| **j** |  |  |
| **k** | **0010** | **01101011** |
| **l** | **0110** | **01101100** |
| **m** |  |  |
| **n** | **101** | **01101110** |
| **o** | **010** | **01101111** |
| **p** | **11111** | **01110000** |
| **q** |  |  |
| **r** | **1101** | **01110010** |
| **s** | **00110** | **01110011** |
| **t** | **111100** | **01110100** |
| **u** | **11000** | **01110101** |
| **v** |  |  |
| **w** |  |  |
| **x** |  |  |
| **y** | **0001** | **01111001** |
| **z** |  |  |

**Question 2**



Bar tending is no better (excluding space)= 1100111101101 1111000000101001111001010111 10000110 101010 11001000011110011110000001101

**Do you think that the Huffman codes obtained are unique? If not, can you think of any other equivalent codes that would give the same compression? Answer:**

They are unique but there are several alternative to this table for the same input. For instance, the frequency of e and y is 2 and same length so we can swap them. If we swap e and y, we can get an alternative Huffman code for the exact same input. ­­­­

**Question 3**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Frequencies from Document 1** | **Frequencies from Document 2** | **short sentence** | **Letter** | **Relative Frequencies Document 1** | **Relative Frequencies Document 2** | **Relative Frequencies Short sentence** |
| 16480 | 9714 | 4 | a | 0.0937322 | 0.08489923 | 0.081632653 |
| 2553 | 1873 | 1 | b | 0.0145205 | 0.0163698 | 0.020408163 |
| 7073 | 4832 | 2 | c | 0.0402286 | 0.04223112 | 0.040816327 |
| 5760 | 3435 | 1 | d | 0.0327608 | 0.0300215 | 0.020408163 |
| 19330 | 13432 | 3 | e | 0.109942 | 0.11739412 | 0.06122449 |
| 5363 | 2938 | 1 | f | 0.0305028 | 0.02567778 | 0.020408163 |
| 2987 | 2837 | 3 | g | 0.016989 | 0.02479505 | 0.06122449 |
| 6304 | 3755 | 1 | h | 0.0358549 | 0.03281826 | 0.020408163 |
| 16442 | 12173 | 3 | I | 0.0935161 | 0.1063906 | 0.06122449 |
| 333 | 212 | 1 | j | 0.001894 | 0.00185286 | 0.020408163 |
| 2444 | 1640 | 1 | k | 0.0139006 | 0.01433341 | 0.020408163 |
| 9110 | 5835 | 2 | l | 0.0518144 | 0.05099722 | 0.040816327 |
| 4570 | 2567 | 2 | m | 0.0259925 | 0.02243528 | 0.040816327 |
| 10890 | 8396 | 3 | n | 0.0619383 | 0.07338006 | 0.06122449 |
| 10266 | 5860 | 3 | o | 0.0583893 | 0.05121572 | 0.06122449 |
| 6211 | 4204 | 2 | p | 0.0353259 | 0.03674247 | 0.040816327 |
| 126 | 92 | 1 | q | 0.0007166 | 0.00080407 | 0.020408163 |
| 11366 | 6583 | 3 | r | 0.0646457 | 0.05753465 | 0.06122449 |
| 10988 | 7075 | 4 | s | 0.0624957 | 0.06183468 | 0.081632653 |
| 14930 | 9573 | 1 | t | 0.0849164 | 0.08366691 | 0.020408163 |
| 3401 | 1889 | 2 | u | 0.0193436 | 0.01650964 | 0.040816327 |
| 1479 | 979 | 1 | v | 0.008412 | 0.00855635 | 0.020408163 |
| 3815 | 2357 | 1 | w | 0.0216983 | 0.02059991 | 0.020408163 |
| 860 | 515 | 1 | x | 0.0048914 | 0.00450104 | 0.020408163 |
| 2419 | 1563 | 1 | y | 0.0137584 | 0.01366044 | 0.020408163 |
| 320 | 89 | 1 | z | 0.00182 | 0.00077785 | 0.020408163 |

The sentence I used is ‘How razorback-jumping frogs can level six piqued gymnasts’

C) I believe it’s a Zipfian distribution.

**How many words were required to get a good estimate of the true frequencies? Answer:**

My small sentence used each letter of alphabet and gave a good estimate for more frequent words but for the less frequent ones the relative frequencies were too high. So, I would assume that a thousand letters would be enough but at the end of the day it all depends on the subject of the article that you are getting the words from.

**Question 4**

|  |  |
| --- | --- |
| **Guess of Most Frequent Letter Pairs** | **Most Frequent Pairs as Computed by the Java Huffman Tree Application** |
| **Th** | **re** |
| **sh** | **an** |
| **is** | **it** |
| **re** | **ti** |
| **ch** | **er** |
| **er** | **ef** |
| **io** | **le** |
| **it** | **as** |
| **he** | **th** |
| **te** | **st** |

|  |  |
| --- | --- |
| **Average Compression using Single Letters** | **46.9** |
| **Average Compression using Letter-Pairs** | **50.8** |
| **Average Improvement in Compression** | **3.9** |

**Can we expect to get ever better compression with increasing size of the combination? Answer:**

We might assume that by increasing the number of letters while compressing the compression percentage would be higher, but we should also remember that this java program doesn’t factor the fact that there might be spaces between letters.