Mark Kazzaz

2018-11-30

Bellevue College, CS211, Fall Quarter

Assignment 8 – Chapter 17

|  |  |
| --- | --- |
| **Overall**: Not all requirements have been met. Code can build a HuffmanNode, HuffmanTree, character mapping, priority queue of nodes, and compress text into a HuffmanTree. Code cannot decompress or correctly show the individual bytes in the output stream dialog box. Given I’m approaching the deadline, I am submitting what I have knowing it is not complete. | |
| **Requirement** | **Evidence via debugging screenshots** |
| Generate a map of <Char, Int> by reading a FileInputStream | Start of getCounts(), empty mapping, first byte read into readInt variable:    Finished reading the FileInputStream, fully populated character mapping, keepRunning variable causes WHILE loop to end: |

|  |  |
| --- | --- |
| Generate a priority queue of HuffmanNode type based on character mapping | Priority queue initialized, character mapping available as charCounts, FOR LOOP starting:    End of FOR LOOP, priority queue size now matching charCount mapping, contains HuffmanNode type objects: |

|  |  |
| --- | --- |
| Build a HuffmanTree | Start of buildHuffmanTree(), charCounts map available, priority queue constructed:    When queue size is > 1; two nodes are popped out and joined into a parent node:    Loop continues until queue contains only a single node, then overallRoot is assigned, expanding on children shows HuffmanTree has been built: |

|  |  |
| --- | --- |
| Print visual of tree via sideways visual | GUI calls printSideways() and recursively populates a string representing the binary tree. GUI presents final tree in a dialog box: |