Checkpoint.pdf

1. Describe how you built the tree and how you find the codeword for each byte.

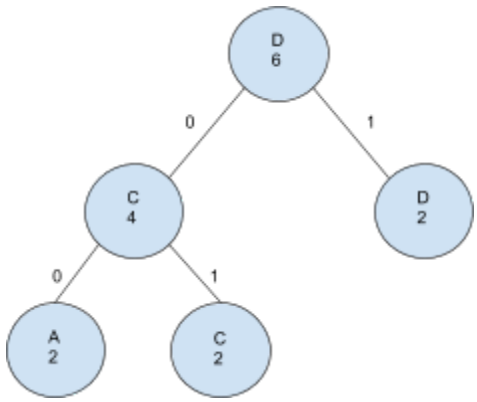
In order to create the Huffman tree, first we create a node for each distinct character (i.e. A, B, C) from the possible 256 ASCII characters that show up in the file to encode. Next we assign that node a data field "count" that represents how many times that character appears in the file we are attempting to encode. We now have a field of nodes with their respective frequencies. We take the two nodes with the lowest frequencies and create a new node with a 0 child (the alphabetically larger node) and a 1 child (the alphabetically smaller node). This new node is has the sum of the frequencies of its two children and is now a viable node to choose from in the forest of tree nodes. This process is repeated until we are left with one tree node left and this is the root of our tree.

Our tree is now constructed and in order to find the codeword for a specific ASCII character we traverse down the tree path to the specified node. We record this nodes path of 0s and 1s to find the codeword for the character.

2.
Checkpoint1.txt -> AADDCC
Checkpoint2.txt-> LOUIS

Compressors output for 1:
0
0
…
2
0
2
2
0
…
0
0011111010

Handwritten:



0000110101

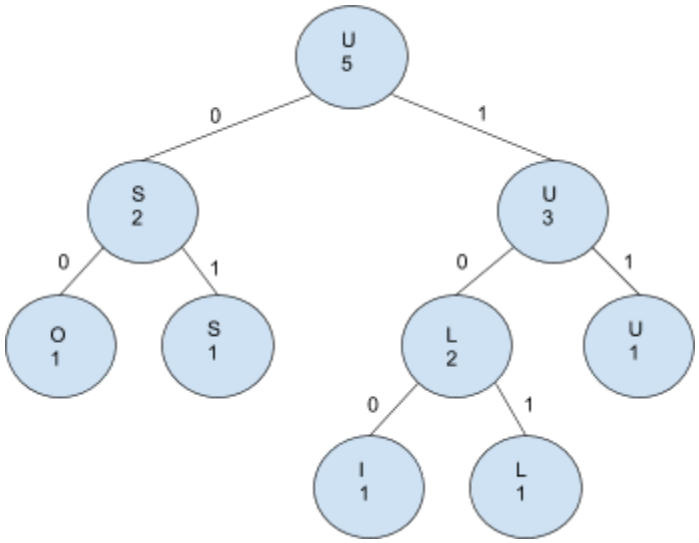
- Output did not match, changed our comparator method from < to > and the output matched.

Compressors output for 2:

0
0
…
0
1
0
0
1
0
0
1
0
0
0
1
0
1
0
…
0

101001110001

Handwritten:



101001110001

- Outputs match