Algorithmics I, 2021 - Assessed Exercise Mark Sheet

NAME: Karlis Siders 2467273

Huffman implementation, correctness and efficiency (10 marks): **9**

A better choice for storing the frequencies would be an array of size 128 using the ascii encodings. This would be more efficient for updating frequencies as you read through the text. There is a large overhead using a hashmap. Well done on not actually building the tree but just simulating the steps and finding the WPL, and hence the size of the compressed file. Also a priority queue is a good choice. Overall an excellent solution well done.

LZW implementation, correctness and efficiency (10 marks): **7**

You have not followed the logic of the algorithm which is find longest string output code word for that string and only after that insert the new string updating the codeword length if needed, you are adding and updating the codeword length first. So you are updating the codeword length a step too early. When reading the file you need to use hasNextLine rather than hasNext as this ignores the lines at the end of the file that are empty. After this change your results are correct. Also there is always a last string to encode. After making these changes your code works. It was just luck these errors meant you got the right result so no need for you to think they were there. Search is efficient keeping track of the current node, but you can use the same thing for insert and just add a child to the last node found during search.

Report, quality of submitted code, overall presentation (8 marks): **6**

Good report but there could have been more on the data structures chosen and efficiency considerations. Code is clear and well structured. Had to remove package commands to compile.

Outputs from acceptance test (2 marks): **2**

Included in report.

TOTAL (30 marks): 24