README

I worked with my partner Shashank Adepu on the Huffman Encoding Project. The parts I worked on are as follows.

I used a BitInputStream object to read bits from a file into an arrayList named ‘values’. Thus, every character from the file was read as an ASCII value into the arrayList. I went through arrayList ‘values’ to identify duplicates and maintained two lists of the ‘keys’ and ‘freq’ (containing the frequency of occurrence of a particular key in the file that was read). I used insertion sort to sort the two lists by increasing order of frequency. Using the TreeNode class provided in the assignment description, I recursively built a huffman tree starting from the leaves (representing the lowest frequencies) to the root (the highest frequency). I stored the values from ‘keys’ arrayList as the value of the node and the values from ‘freq’ arrayList as the weight of the node.

In order to read in a compressed Huffman file, I created a method called ‘readInput()’ that used a BufferedReader to read in all of the bits (1’s and 0’s) in the file. The BufferedReader read input one character at a time. The characters were concatenated to a String and as soon as one encoding was finished, the String was sent to a decompressing method which decoded the String and printed it using PrintWriter to a decompressed file. I made sure to account for the EOF character (-1) so the BufferedReader stopped reading at the end of the file. Each character encoding was also separated by a space so I also used the space to determine when a character encoding was finished.

We ran several different text files on the program, all of which were successfully compressed and decompressed. The uncompressed and decompressed files were stored in the default location (in the directory containing the src and bin folders). We did not consult with any TAs.

Overall, I spent about 14 hours on this assignment. I thought it was very good experience with compression/decompression. I learned quite a bit and it also allowed me to review many common Java concepts such as reading and writing to a file, arrayLists, arrays, and building/searching trees.