CSE 100
1.23.2016
Dadong Jing
Rui Deng
PA 2

For our compression, we first check if the input arguments are valid, and if the inputs are not valid, we print out a warning to show user that they enter invalid inputs. Then, we open the input file and output file, which user has specified. We set up a vector of size 256 to hold the frequency of each characters in the input file. After that, we use this frequency vector to construct a HCTree. Then, we close the input file and reopen it so that the cursor of the input file will be at the start of the input file. We use a for loop to loop through the frequency vector and write to the compress file in the following way: frequency[0], space, frequency[1],space..... We do not need to store the char symbol since the index of the frequency vector corresbodes to the ascii of the characters. Once we have our header, we will call encode method to encode each character based on the frequency. In the encode method, we first find the character's position in the leaves vector, and we use while loop to loop from the bottom to the top and in the process, we record 0 if it is a c0, and 1 if c1. (use a vector to hold thoses 0's and 1's, and the order are from bottom to top) Then we write the recorded 0's and 1's to the compressed file(the order is also from back of the vector to the front so that the encoded message can be decoded from top to bottom). During the ucompress process, we create a vector of size 256, and we read the header from the compressed file and update the vector we just created in order to have a frequency vector which we can use to reconstructe the HCTree. After that, we call the decode method to get the decoded message. In decode method, we read from a file, and if we read a 0, we go to c0 node, and if we read a 1, we go to c1 node. We keep looping until we reach the point that there is no child node. And we write the node's symbol to the uncompressed file to get the decoded message.