Huffman

a. Given the characters ‘a’ to ‘h’ we have 8 unique characters which means the minimum number of bits needed to represent 8 unique values would be log2(8) which is 3 bits.

   'a': 64

   'b': 35

   'c': 79

   'd': 48

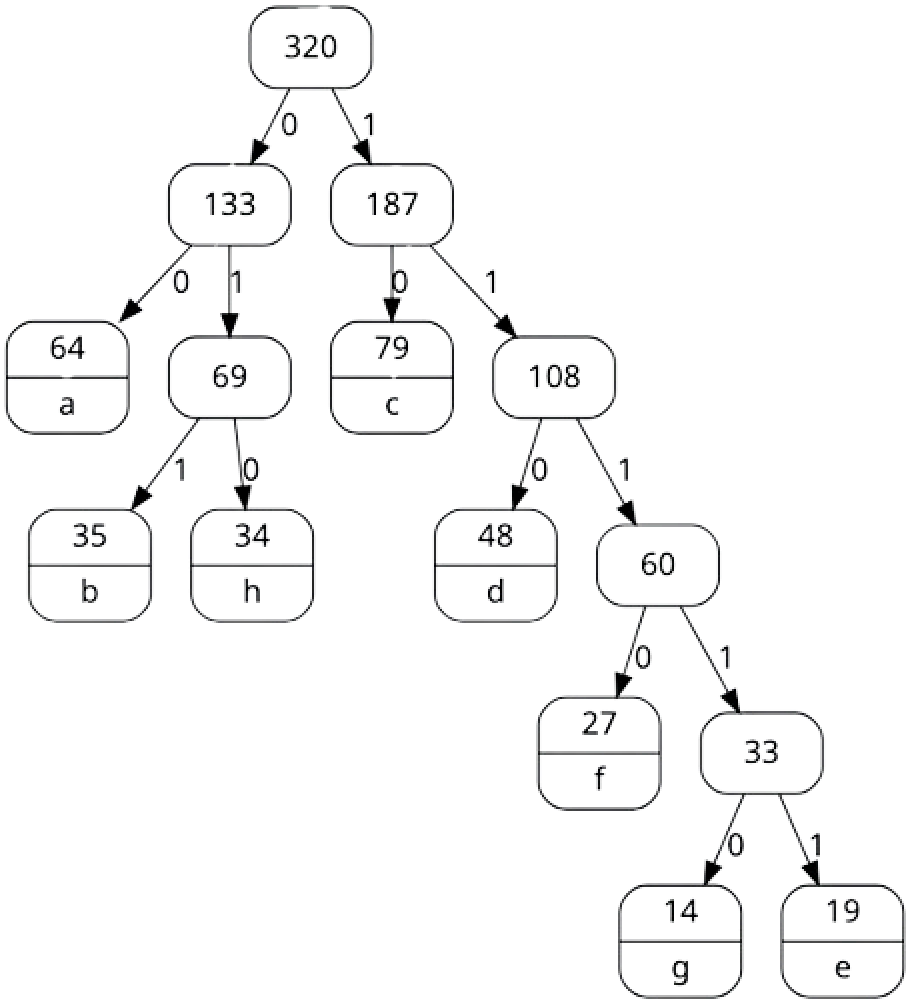
   'e': 19

   'f': 27

   'g': 14

   'h': 34

b.



c.

a': 00

'b': 011

'c': 10

'd': 110

'e': 11111

'f': 1110

'g': 11110

'h': 010

d.

e. 910 bits

f. 0.947

Exercise 16
To find the minimum number of steps required to reach a target integer n from 1, using increments and doublings, work backwards from n. If n is odd, subtract 1; if even, divide by 2. Repeat this until you reach 1, counting each operation as a step. This greedy algorithm ensures the shortest path to n because it reverses the most efficient moves—halving trumps incrementing when possible, which aligns with the operations that lead to n.

Exercise 22a
To verify if a string of parentheses is balanced, you can use an algorithm that iteratively scans the string maintaining a count of open parentheses. For each '(' encountered, the count is increased, and for each ')', the count is decreased. If at any point the count falls below zero, or if it's not zero at the end, the string is unbalanced. For the second part, finding the length of the longest balanced subsequence within a string can be done with a greedy algorithm that counts the number of '(' and ')' characters, only pairing them when both are present, which inherently ensures balanced pairs. Both algorithms operate in linear time, providing efficiency with a complexity of O(n).