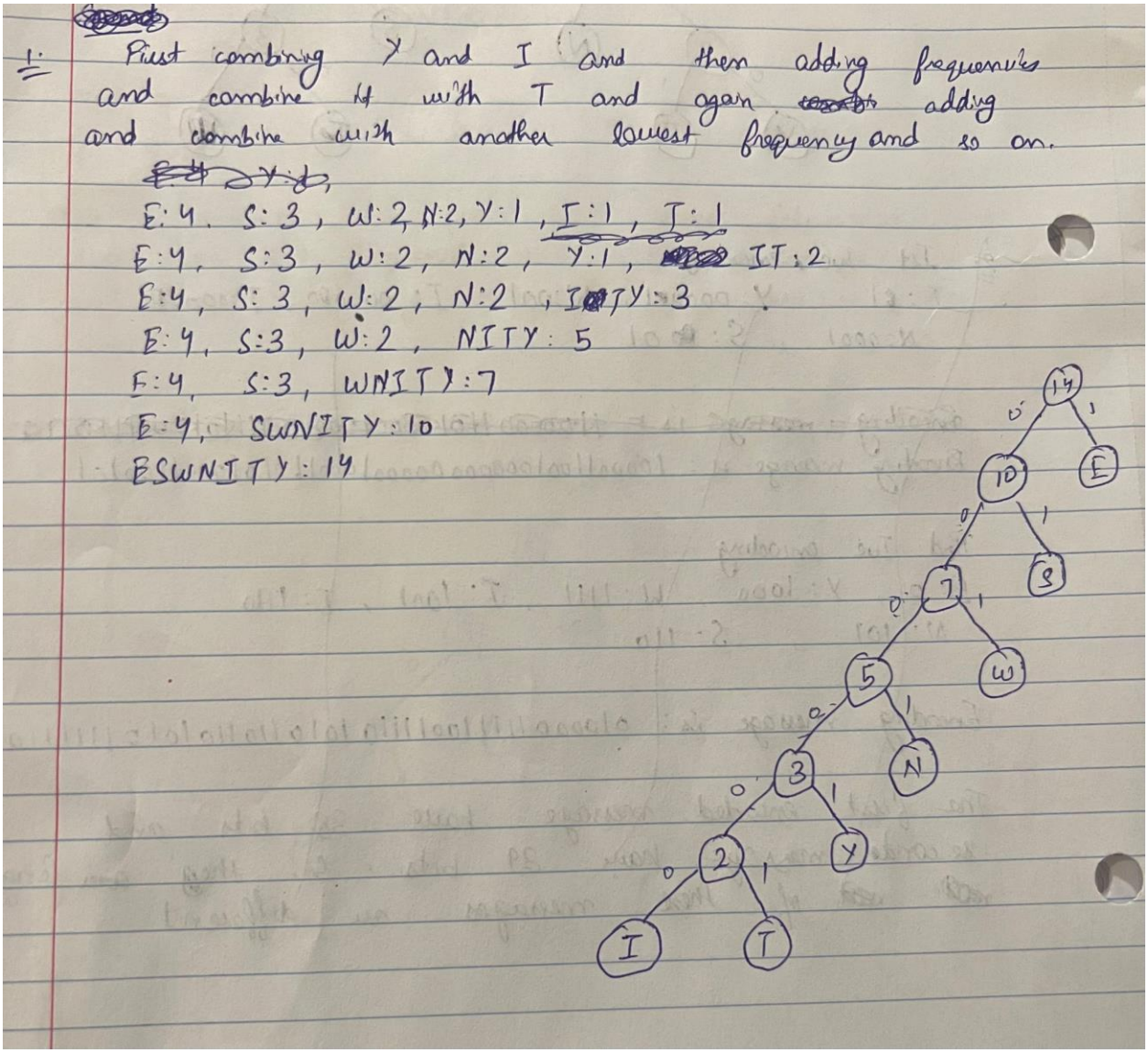
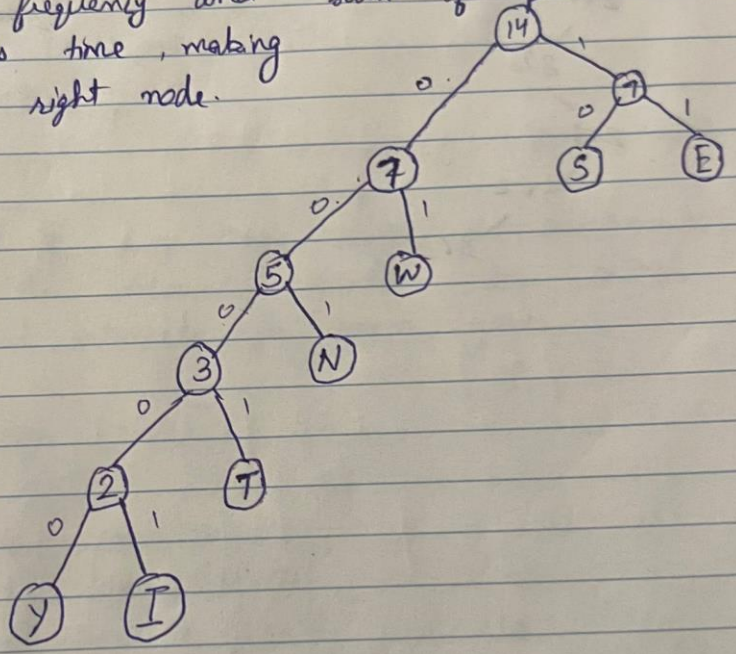
Exercise 1:


1:
First combining Y and I and then adding frequencies and combine it with T and again adding and combine with another lowest frequency and so on.
E:4, S:3, W:2, N:2, Y:1, I:1, T:1
E:4, S:3, W:2, N:2, Y:1, IT:2
E:4, S:3, W:2, N:2, ITY:3
E:4, S:3, W:2, NITY:5
E:4, S:3, WNITY:7
E:4, SWNITY:10
ESWNITY:14
14
0
1
10
E
0
1
7
S
0
1
5
W
0
1
3
N
0
1
2
Y
0
1
I
T

Frequencies :-

E: 4 , Y: 1 , W: 2 , I: 1 , T: 1 , N: 2 , S: 3

Q: Again combining two nodes with lowest frequencies, Y and I and sum that frequencies. and again combine one with lowest frequency and sum of previous two trees and this time, making N and S as right node.



Q: Ist.

E: 11 , Y: 00000 , W: 01 , I: 00001 , T: 0001 , N: 001
S: 10

Message: 11000001101000100011001111010001110

IInd.

E: 1 , Y: 00001 , W: 001 , I: 000000 , T: 000001 , N: 0001 ,
S: 01

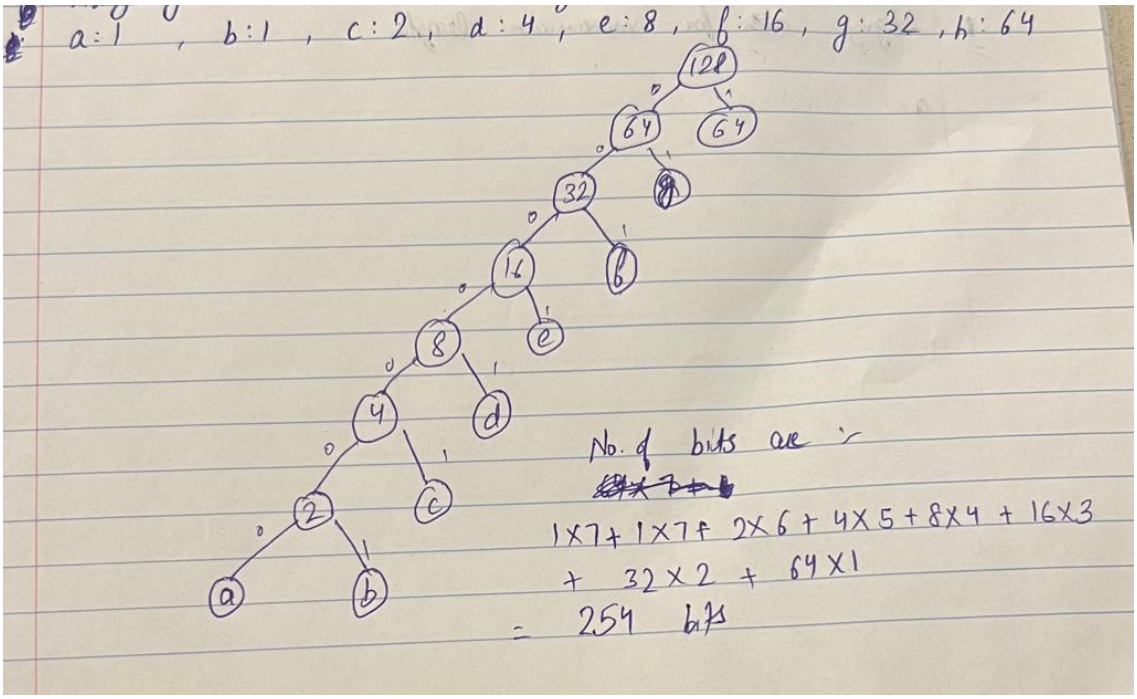
Message: 1000011001000000000001000110101000110010 1

Ist message have 38 bits and 2nd have 41 bits. They are of different size.

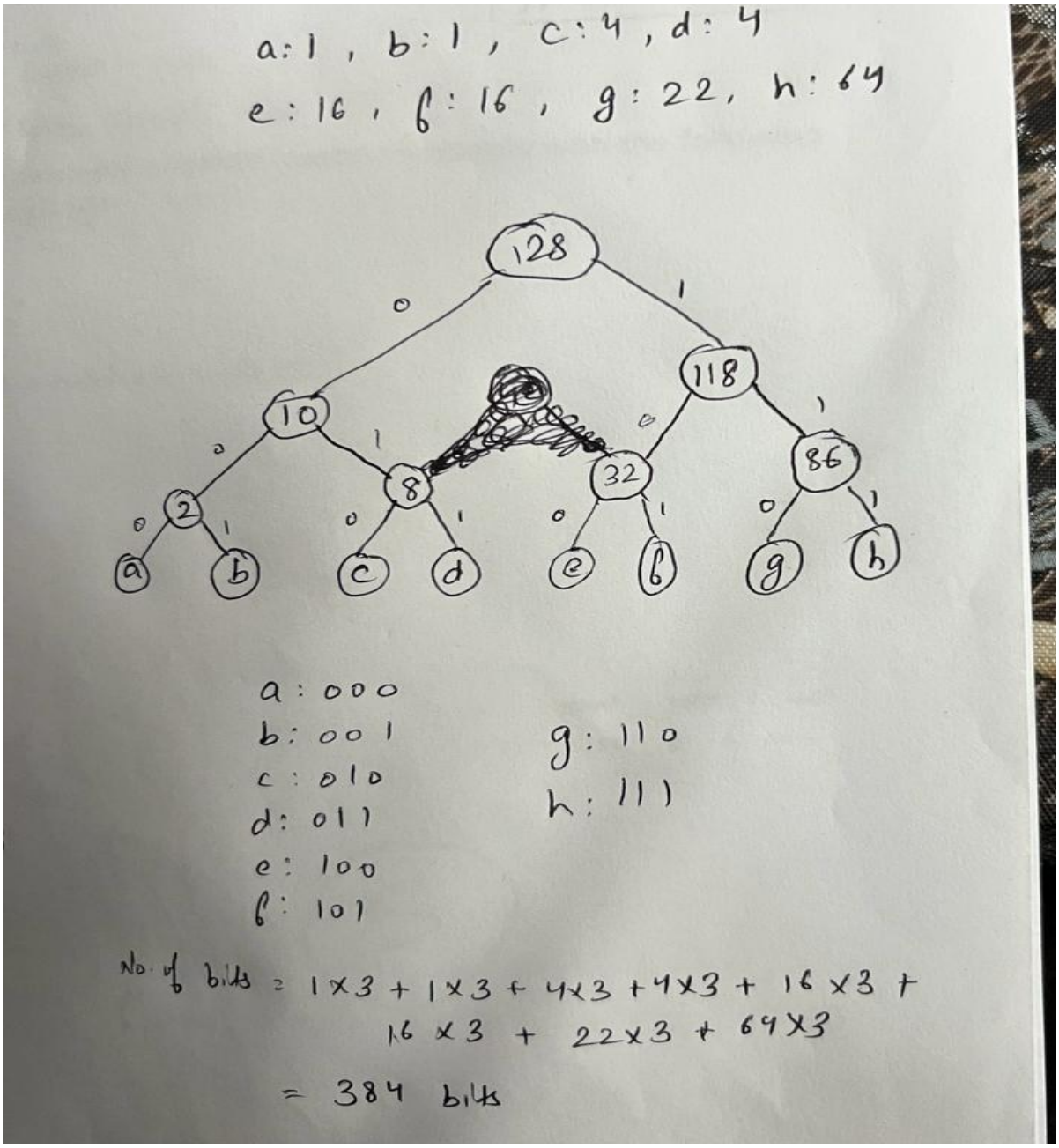
3. Yes, the length depends on our choice in building the trie. The best encoding tree depends on the frequencies and the compression ratio of the message. If the frequency is high and node that deepest node. It use lower bits as compared to others.

Exercise 2:

1.For max height



2.For min height



Exercise 4:

Complexity for makeTree()

Here First code runs n times to get frequencies and another for loop again runs n times to pushing the nodes.

While loop runs until queue is not 1. Each time it pop 2 elements and push one into the node, it make complexity logn.

Overall complexity is nlogn.

Complexity for print()

Time complexity is n; because it goes through each node in a tree and print that node if its left and right node is nullptr.