Huffman Encoding

What if we use shorter codes for frequent characters and longer codes for infrequent characters?

for example:
e: 01
q: 00110110111011110

How should we do this?

1. Build a Huffman Tree
2. Traverse the Huffman Tree and assign codes to characters.

1. Building a Huffman Tree

Input: Array of unique characters along with their frequency of occurrences

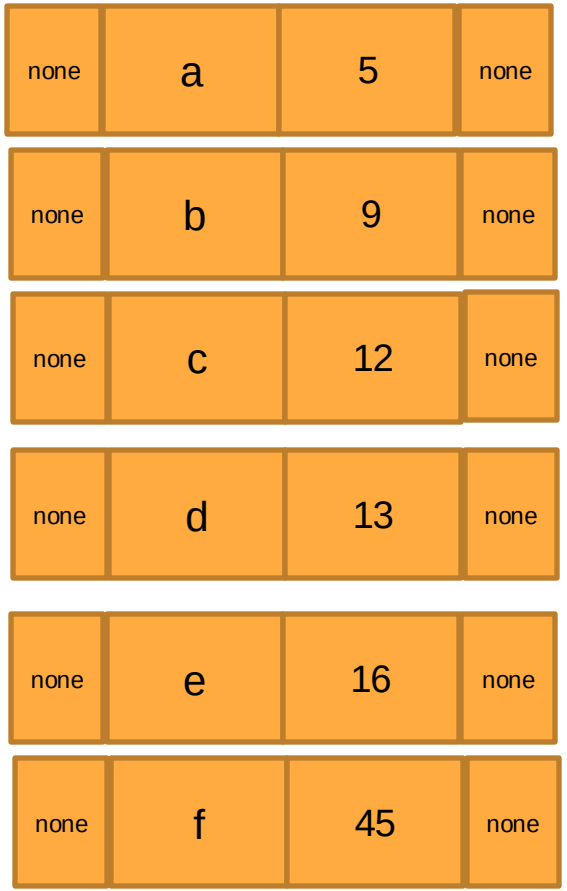
Character	Frequency
a	5
b	9
c	12
d	13
e	16
f	45

Output: Huffman Tree

Create TreeNodes

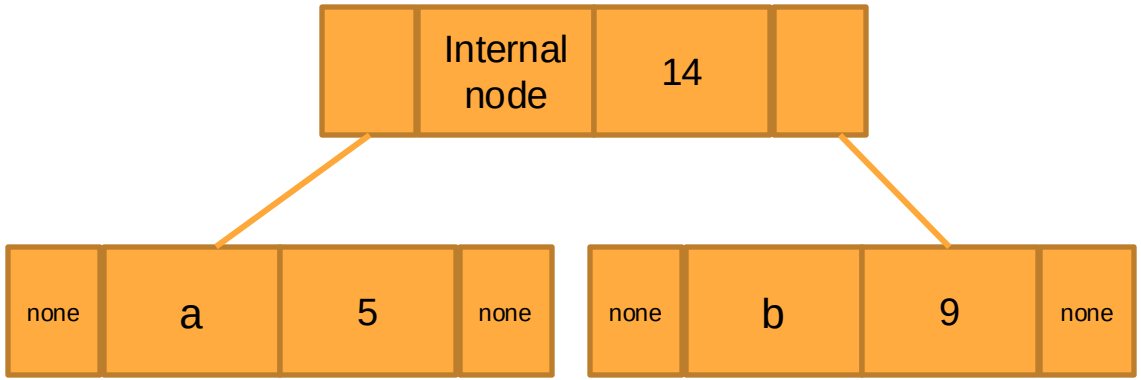
- Data = character
- Key = Frequency

Character	Frequency
a	5
b	9
c	12
d	13
e	16
f	45



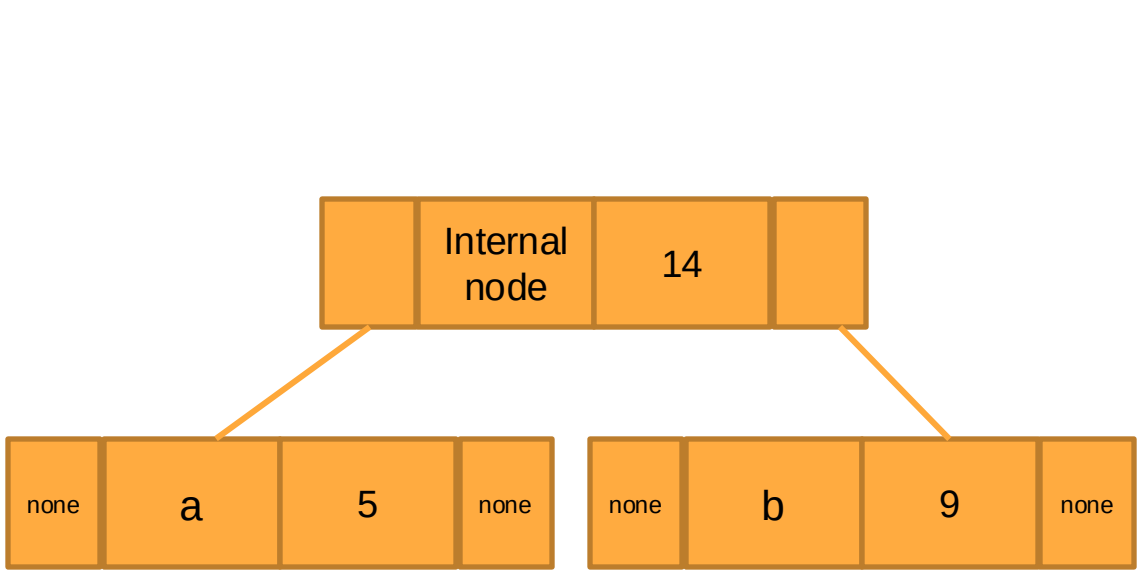
Extract 2 minimum frequency nodes

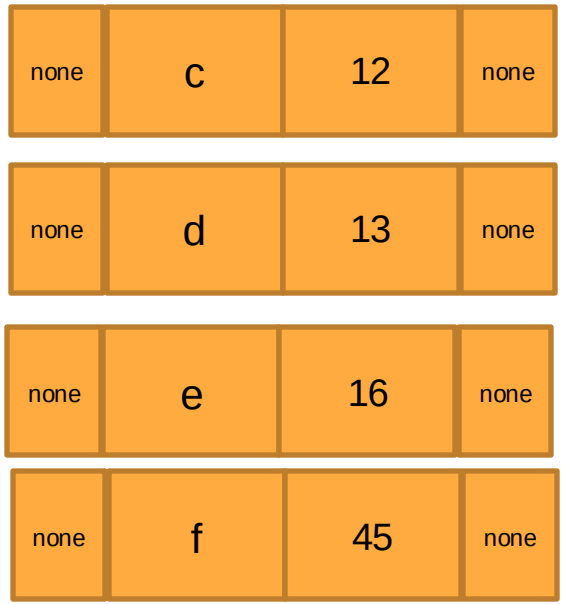
Create a new internal node with frequency equal to the sum of the two nodes frequencies. Make the first extracted node as its left child and the other extracted node as its right child.

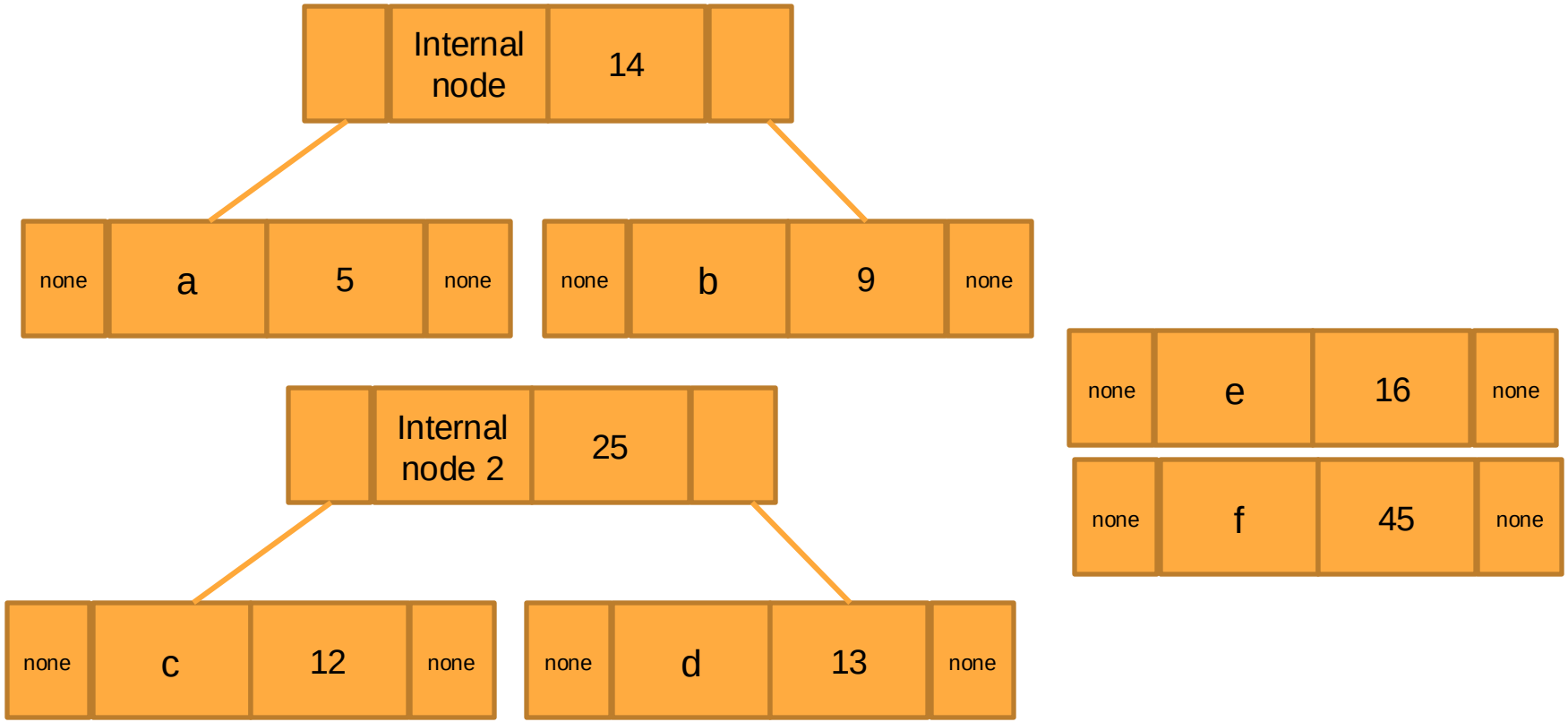


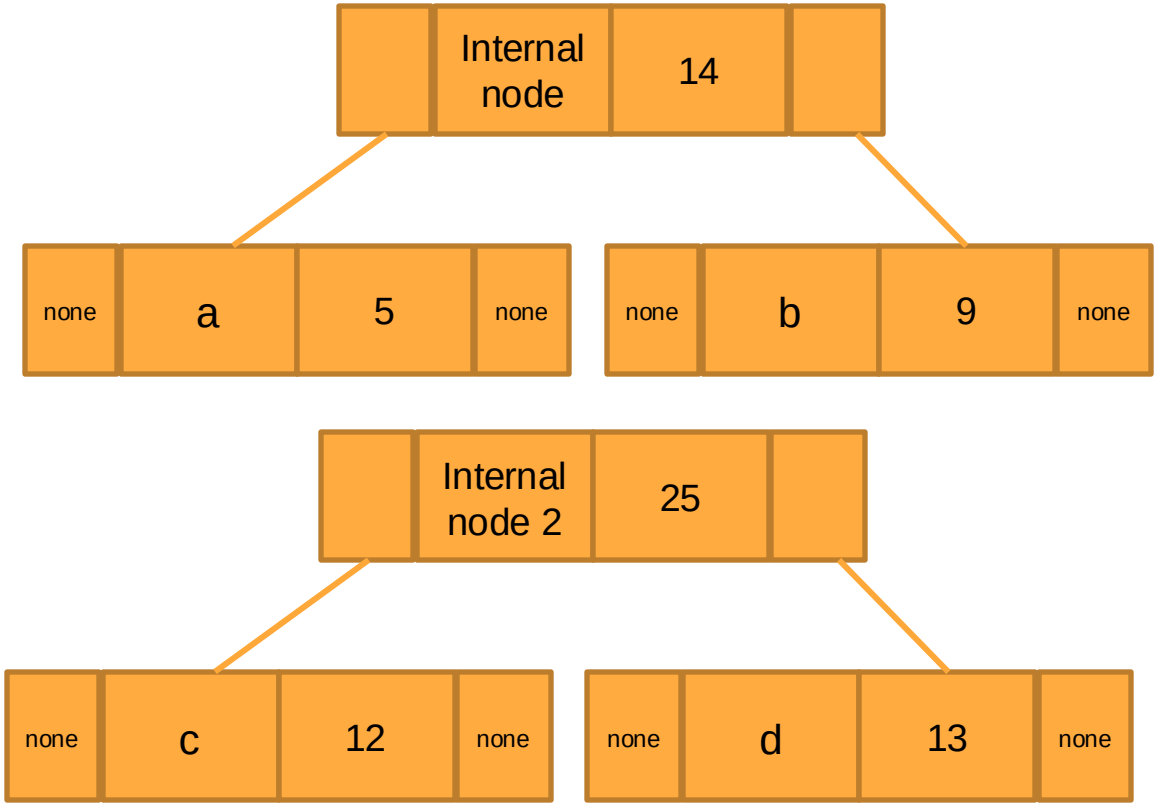
Character	Frequency
c	12
d	13
Internal Node	14
e	16
f	45

Repeat those steps until the tree is complete

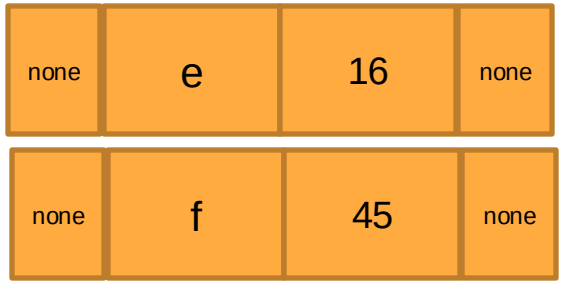


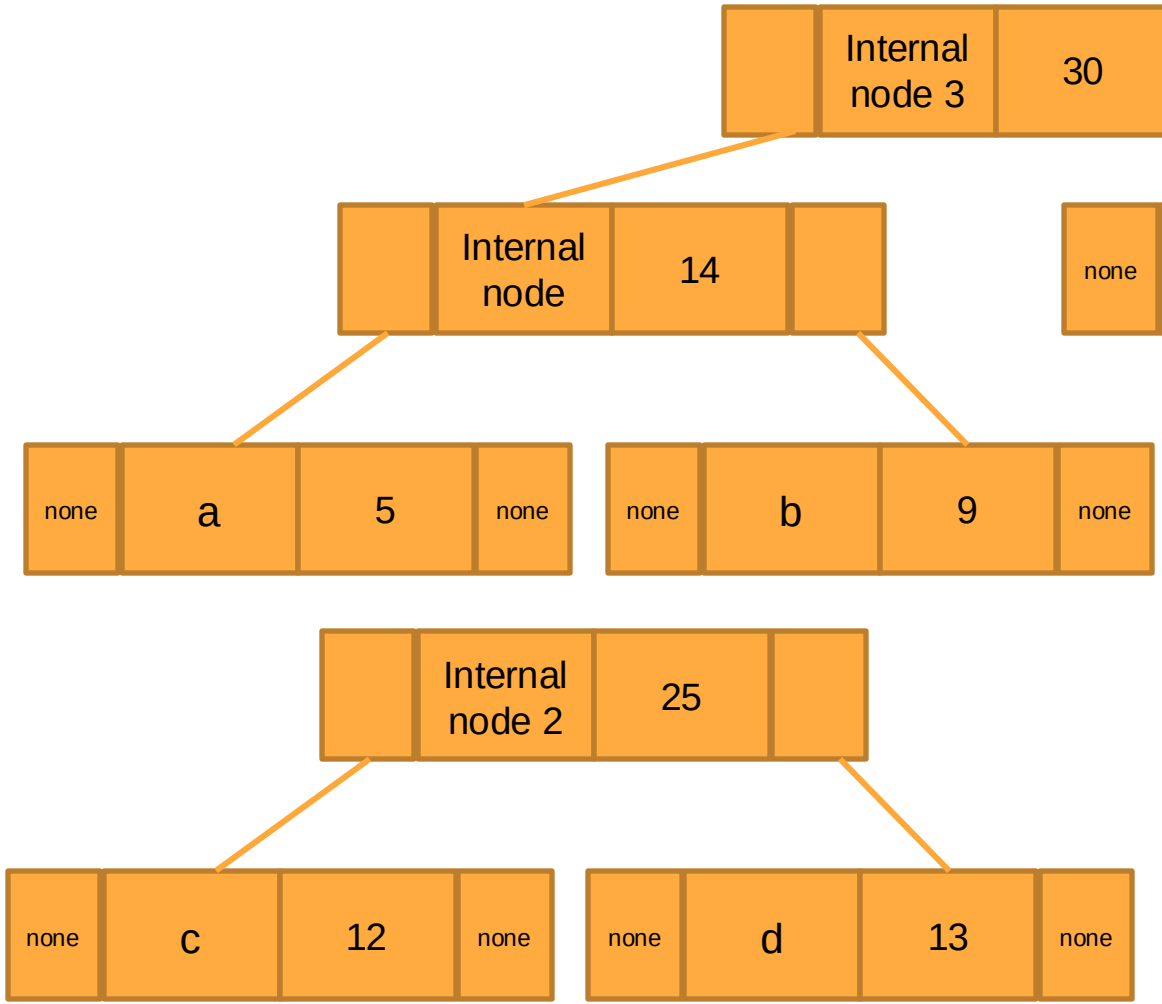



Internal node
14
none
a
5
none
none
b
9
none
Internal node 2
25
none
c
12
none
none
d
13
none
none
e
16
none
none
f
45
none

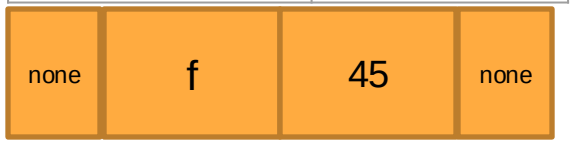


Character	Frequency
Internal Node	14
e	16
Internal Node 2	25
f	45

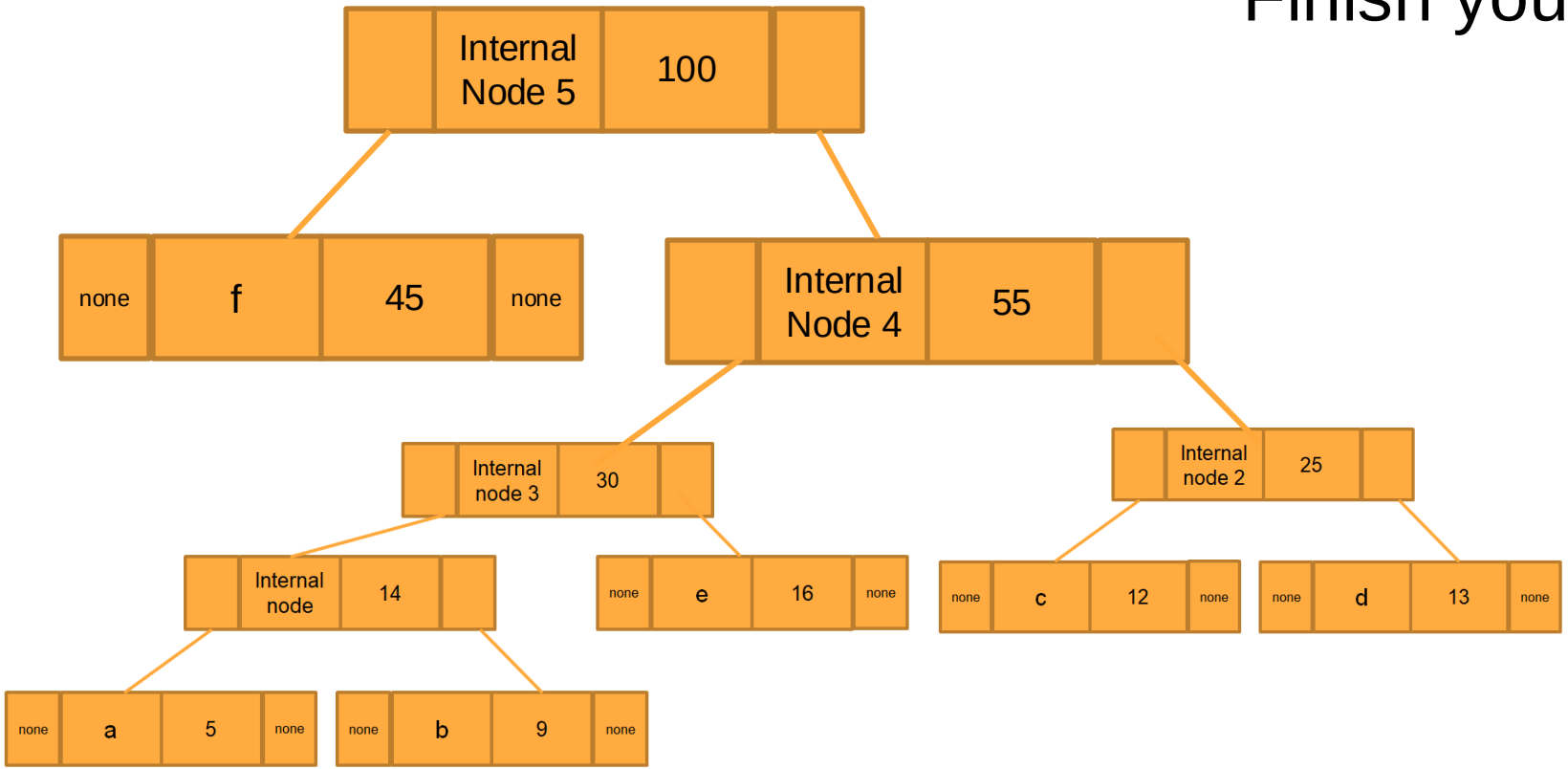


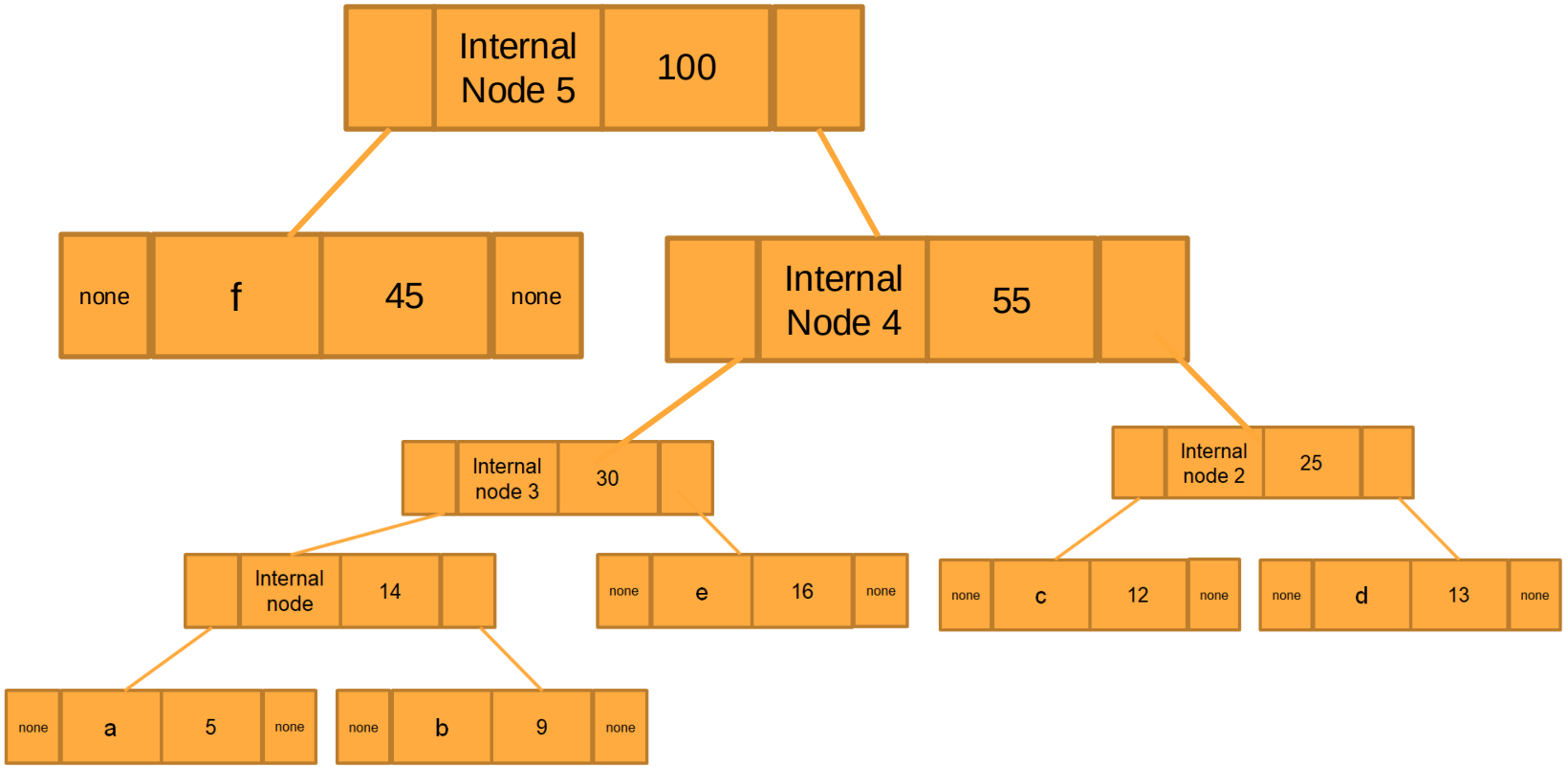


Character	Frequency
Internal Node	14
Internal Node 2	25
Internal Node 3	30
f	45



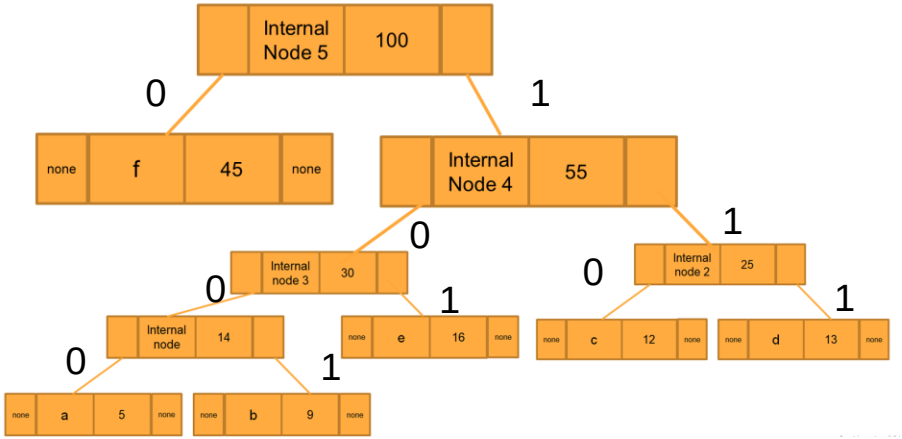
Finish yourself!




Internal Node 5
100
none
f
45
none
Internal Node 4
55
Internal node 3
30
Internal node 2
25
Internal node
14
none
e
16
none
none
c
12
none
none
d
13
none
none
a
5
none
none
b
9
none

2. Traverse the Huffman Tree and assign codes to characters

Traverse the tree formed starting from the root. While moving to the left child, write 0 to the array. While moving to the right child, write 1 to the array. Print the array when a leaf node is encountered.



Character	Code Word
f	0
c	100
d	101
a	1100
b	1101
e	111