Code:

```python
import heapq
class Node:
    def __init__(self, freq, symbol, left=None, right=None):
        self.freq = freq
        self.symbol = symbol
        self.left = left
        self.right = right
        self.huff = ""

    def __lt__(self, nxt):
        return self.freq < nxt.freq

def printNodes(node, val=""):
    newVal = val + str(node.huff)
    if node.left:
        printNodes(node.left, newVal)
    if node.right:
        printNodes(node.right, newVal)
    if not node.left and not node.right:
        print(f"{node.symbol} -> {newVal}")

chars = ["a", "b", "c", "d", "e", "f"]
freq = [5, 9, 12, 13, 16, 45]
nodes = []
for x in range(len(chars)):
    heapq.heappush(nodes, Node(freq[x], chars[x]))
while len(nodes) > 1:
    left = heapq.heappop(nodes)
    right = heapq.heappop(nodes)
    left.huff = 0
    right.huff = 1
    newNode = Node(left.freq + right.freq, left.symbol + right.symbol, left, right)
    heapq.heappush(nodes, newNode)

print("Huffman Tree : ")
printNodes(nodes[0])
```

Output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements!
https://aka.ms/PSWindows

PS E:\BE\41427_LP-III_Codes\DAA> & C:/Users/abhij/AppData/Local/P
rograms/Python/Python311/python.exe e:/BE/41427_LP-III_Codes/DAA/
41427_LP-III_A2.py
Huffman Tree :
f -> 0
c -> 100
d -> 101
a -> 1100
b -> 1101
e -> 111
PS E:\BE\41427_LP-III_Codes\DAA>
```