

92. optimal tree problem: Huffman trees and code

AIM: To solve the optimal tree problem by using the Huffman tree and codes

PROGRAM:

```
import heapq
from collections import defaultdict

def huffman_tree(frequencies):
    heap = [[weight, [symbol, ""]] for symbol, weight in frequencies.items()]
    heapq.heapify(heap)
    while len(heap) > 1:
        lo = heapq.heappop(heap)
        hi = heapq.heappop(heap)
        for pair in lo[1:]:
            pair[1] = '0' + pair[1]
        for pair in hi[1:]:
            pair[1] = '1' + pair[1]
        heapq.heappush(heap, [lo[0] + hi[0]] + lo[1:] + hi[1:])
    return heap[0][1:]

def huffman_codes(tree):
    return {symbol: code for symbol, code in tree}

frequencies = {'a': 45, 'b': 13, 'c': 12, 'd': 16, 'e': 9, 'f': 5}
tree = huffman_tree(frequencies)
codes = huffman_codes(tree)

for symbol, code in codes.items():
    print(f"Symbol: {symbol}, Code: {code}")
```

```
Symbol: a, Code: 0  
Symbol: c, Code: 100  
Symbol: b, Code: 101  
Symbol: f, Code: 1100  
Symbol: e, Code: 1101  
Symbol: d, Code: 111
```

OUTPUT:

TIME COMPLEXITY: $O(n \log n)$