Huffman Coding

In [1]:
```
import heapq
from collections import Counter


class HuffmanNode:
    def __init__(self, char, freq):
        self.char = char
        self.freq = freq
        self.left = None
        self.right = None

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


def build_huffman_tree(freq_dict):
    priority_queue = [HuffmanNode(char, freq) for char, freq in freq_dict.items()]
    heapq.heapify(priority_queue)

    while len(priority_queue) > 1:
        left = heapq.heappop(priority_queue)
        right = heapq.heappop(priority_queue)

        merged_node = HuffmanNode(None, left.freq + right.freq)
        merged_node.left = left
        merged_node.right = right

        heapq.heappush(priority_queue, merged_node)

    return priority_queue[0]


def build_huffman_codes(node, current_code, huffman_codes):
    if node is None:
        return

    if node.char is not None:
        huffman_codes[node.char] = current_code
        return

    build_huffman_codes(node.left, current_code + '1', huffman_codes)
    build_huffman_codes(node.right, current_code + '0', huffman_codes)



input_filename = "input.txt"
compressed_filename = "encoded.txt"


def get_huffman_codes(input_file):
    try:
        with open(input_file, 'r') as infile:
            data = infile.read()

        freq_dict = dict(Counter(data))
        huffman_tree = build_huffman_tree(freq_dict)
        huffman_codes = {}
        build_huffman_codes(huffman_tree, '', huffman_codes)
```

```
        return huffman_codes

    except FileNotFoundError:
        print("Input file not found.")
        return None
    except Exception as e:
        print("An error occurred:", str(e))
        return None

# Example usage:
input_filename = "input.txt"  # Replace with your input file name

huffman_codes = get_huffman_codes(input_filename)
if huffman_codes:
    print("Character - Huffman Code:")
    for char, code in huffman_codes.items():
        print(f"'{char}' - {code}")
```

```
Character - Huffman Code:
'A' - 11
'B' - 10
'C' - 011
'F' - 010
'E' - 00
```

input.txt: AABABBEEEEFEFC

encoded.txt: 11111011101000000000001000010011