```python
import heapq
from collections import defaultdict

def huffman_encode_file(input_file, output_file):
    with open(input_file, 'r') as f:
        text = f.read()

    def build_huffman_tree(frequencies):
        heap = [[weight, [char, ""]] for char, 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 sorted(heapq.heappop(heap)[1:], key=lambda p: (len(p[-1]), p))

    frequencies = defaultdict(int)
    for char in text:
        frequencies[char] += 1

    huffman_tree = build_huffman_tree(frequencies)
    huffman_dict = {char: code for char, code in huffman_tree}

    encoded_text = "".join(huffman_dict[char] for char in text)

    with open(output_file, 'w') as f:
        f.write(encoded_text)

# Usage
input_file = 'input.txt'
output_file = 'huffman_encoded.txt'
huffman_encode_file(input_file, output_file)
```