

## 1 Kodowanie Huffmana - kodowanie.

```
Start
Wczytaj wiadomość message.
occurrences ← pusty słownik
Dla każdej litery w message jako letter wykonuj
|   Jeżeli letter nie występuje w occurrences to
|   |   occurrences[letter] ← 0
|   koniec
|   occurrences[letter] ← occurrences[letter] + 1
koniec
symbols ← pusta kolejka priorytetowa
Dla każdego klucza w occurrences jako letter wykonuj
|   Zakolejkuj trójkę (letter, brak, brak) o wadze  $-occurrences[letter]$  w symbols
koniec
Gdy długość symbols > 1 wykonuj
|   right_node ← Zdejmij element z kolejki symbols
|   right_weight ← Waga elementu right_node w kolejce symbols
|   left_node ← Zdejmij element z kolejki symbols
|   left_weight ← Waga elementu left_node w kolejce symbols
|   Zakolejkuj trójkę (brak, left_node, right_node) o wadze
|    $left\_weight + right\_weight$  w symbols
koniec
root ← Zdejmij element z kolejki symbols
bfs_queue ← pusta kolejka
Zakolejkuj parę (root, "") w bfs_queue
coding_table ← pusty słownik
Gdy długość bfs_queue > 0 wykonuj
|   pair ← Zdejmij element z kolejki bfs_queue
|   node ← pair[0]
|   code ← pair[1]
|   Jeżeli node[1] ≠ brak to
|   |   Zakolejkuj parę (node[1], code + "0") w bfs_queue
|   koniec
|   Jeżeli node[2] ≠ brak to
|   |   Zakolejkuj parę (node[2], code + "1") w bfs_queue
|   koniec
|   Jeżeli node[1] = brak oraz node[2] = brak to
|   |   coding_table[node[0]] ← code
|   koniec
koniec
encoded_message ← ""
Dla każdej litery w message jako letter wykonuj
|   encoded_message ← encoded_message + coding_table[letter]
koniec
Wypisz encoded_message
Stop
```

## 2 Kodowanie Huffmana - dekodowanie.

```
Start
Wczytaj wiadomość encoded_message i tablicę kodową code_table.
decoded_message ← ""
current_sequence ← ""
Dla każdej litery w encoded_message jako letter wykonuj
|   current_sequence ← current_sequence + letter
|   Jeżeli current_sequence występuje w code_table to
|   |   decoded_message ← decoded_message + code_table[current_sequence]
|   |   current_sequence ← ""
|   koniec
koniec
Wypisz decoded_message
Stop
```