

# Wstęp do programowania w Języku C

Krystian Baćłowski

4 grudnia 2017

## Lista 8, Zadanie 2 (15 punktów)

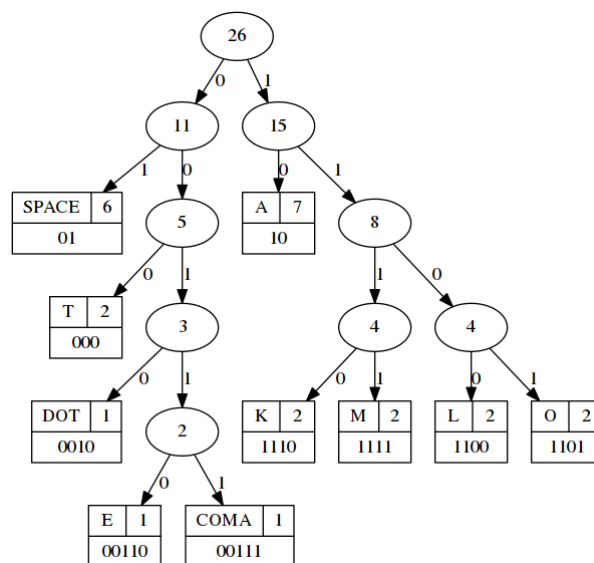
Na kod Morse'a można popatrzeć jak na zapis binarny pewnego alfabetu przy pomocy zer (krótki sygnał) i jedynek (długi sygnał). Każdy kod został dobrany tak, aby odzwierciedlał częstość występowania litery, cyfry lub znaku interpunkcyjnego w tekstach w języku angielskim. Jest to prosty pomysł na minimalizację liczby impulsów niezbędnych do przesłania komunikatu.

Rozwinięciem tej idei w informatyce jest **kodowanie Huffmana**<sup>1</sup>. Dla podanego komunikatu  $M$  nad danym alfabetem (w naszym przypadku ASCII) wyznaczamy częstość występowania znaków i w zależności od tego przypisujemy im kody binarne o minimalnej długości. Kompresja komunikatu  $M$  polega na przetłumaczeniu go na kolejne ciągi bitów odpowiadających oryginalnym literom. Jest to kodowanie dające krótsze komunikaty od kodu Morse'a. Niestety należy przesłać drugiej stronie również tabelę częstości znaków by wiedziała jak odtworzyć kody.

Napisz program, który konstruuje drzewo Huffmana dla tekstu podanego na standardowym wejściu. Na wyjściu standardowym ma pojawić się jedna liczba oznaczająca liczbę bitów, w których zostałby zakodowany podany tekst.

Drzewo Huffmana w liściach przechowuje znaki tekstu wraz częstością ich występowania, a w węzłach wewnętrznych sumę częstości wszystkich znaków w tym poddrzewie. Kod znaku otrzymujemy przechodząc po krawędziach od korzenia do odpowiedniego liścia. Algorytm tworzenia drzewa przystępnie i w detalach wyjaśniono na stronie Wikipedii.

Drzewo Huffmana dla tekstu „ALA MA KOTA, A KOT MA ALE.” może wyglądać jak na poniższym obrazku. Liczba bitów niezbędna do zakodowania naszego tekstu to  $(6 + 7) * 2 + 2 * 3 + (1 + 2 + 2 + 2 + 2) * 4 + (1 + 1) * 5 = 78$  bitów, gdzie oryginalny komunikat miał  $26 * 8 = 208$  bitów.



<sup>1</sup>[https://pl.wikipedia.org/wiki/Kodowanie\\_Huffmana](https://pl.wikipedia.org/wiki/Kodowanie_Huffmana)