Algorytm Huffmana - przykład zlgorytmu zachłannego

Definicja *Drzewo kodu* jest to drzewo binarne, w którego liściach znajdują sie znaki kodowanego tekstu i którego krawędzie są etykietowane bitami 0 lub 1 (krawędź w lewo – 0, w prawo – 1). Kodem znaku jest ciąg bitów na ścieżce od korzenia do liścia zawierającego dany znak.

Długość zakodowanego tekstu (liczona w bitach) przy użyciu kodu reprezentowanego przez drzewo T:

$$B(T) = \Sigma_{z \in C} z.f \cdot d_T(z)$$

gdzie

> C to zbiór znaków
>
> $z.f$ to ilość wystąpień znaku z w kodowanym tekście
>
> $d_T(z)$ to głębokość w drzewie T liścia zawierającego znak z, czyli długość kodu znaku z

Drzewo kodu o najmniejszym $B(T)$ będziemy nazywać *optymalnym drzewem kodu*

Uwaga. Może istnieć wiele optymalnych drzew kodu dla danego tekstu.

Stwierdzenie 1 Optymalne drzewo kodu jest drzewem regularnym, czyli każdy węzeł ma albo 2 synów albo 0.

Stwierdzenie 2 Niech $x, y \in C$ będą znakami o najmniejszych ilościach wystąpień w kodowanym tekście. Istnieje optymalne drzewo kodu, w którym x i y są braćmi na najniższym poziomie drzewa.

Stwierdzenie 3 Niech T będzie drzewem kodu w którym x i y są liśćmi braćmi a z ich ojcem. Niech drzewo T' będzie drzewem otrzymanym z T przez usunięcie liści x i y i przypisanie znakowi z (który teraz stał się liściem) ilości wystąpień $z.f = x.f + y.f$. Wtedy

$$B(T) = B(T') + x.f + y.f$$

`x,y,z` – węzły drzewa kodu
`x.f` – ilość wystąpień (pseudo)litery zawartej w `x`

```
Huffman(C)
  zbuduj kolejkę Q zawierającą litery alfabetu C
  for i = 2 to |C| do
    x = ExtractMin(Q)
    y = ExtractMin(Q)
    utwórz nowy węzeł z
    z.left = x
    z.right = y
    z.f = x.f+y.f
    Insert(Q,z)
  return ExtractMin(Q)
```

Stwierdzenie Ze stwierdzeń 2 i 3 wynika, że algorytm Huffmana znajduje optymalne drzewo kodu.

Stwierdzenie Złożómość pesymistyczna algorytmu Huffmana dla alfabetu zawierającago n znaków jest:

> $\Theta(n\, lg\, n)$, jeżeli kolejka priorytetowa używana w algorytmie jest implementowana jako kopiec binarny
>
> $\Theta(n^2)$, jeżeli kolejka priorytetowa jest implementowana jako tablica lub lista