

PROGRAM 01

Użyj algorytmu Huffmana w celu skonstruowania optymalnego binarnego kodu prefiksowego dla liter przedstawionych w poniższych tabelach:

a)

| Litera | Częstotliwość |
|--------|---------------|
| A | 12 |
| B | 7 |
| I | 18 |
| M | 10 |
| S | 9 |
| X | 5 |
| Z | 2 |

b)

| Litera | Prawdopodobieństwo |
|--------|--------------------|
| c | 0,11 |
| e | 0,22 |
| i | 0,16 |
| r | 0,12 |
| s | 0,15 |
| t | 0,10 |
| x | 0,14 |

OPIS ALGORYTMU HUFFMANA

Huffman()

*for każdego symbol utwórz drzewo zawierające jedynie korzeń
i uporządkuj te drzewa na podstawie prawdopodobieństwa
wystąpienia poszczególnych symboli;*

*while pozostało więcej niż jedno drzewo
weź dwa drzewa t1 i t2 o najmniejszych prawdopodobieństwach
p1 i p2 ($p1 \leq p2$) i utwórz drzewo z dziećmi t1 i t2,
a w jego korzeniu zapisz prawdopodobieństwo $p1+p2$;*

Z każdej lewej gałęzi skojarz 0, a z prawą gałęzi skojarz 1;

Dla każdego symbolu utwórz unikalne słowo kodowe, przechodząc drzewo od korzenia do węzła zawierającego prawdopodobieństwo odpowiadające danemu symbolowi i łącząc ze sobą wszystkie wystąpienia zer i jedynek;

PRZYKŁADY (LABORATORIUM)

Przykład 1

Założmy, że zbiór znaków ma postać {a, b, c, d, e, f} i każdy znak występuje w pliku pewną liczbę razy, określoną w poniższej tabeli:

| Litera | Częstotliwość |
|--------|---------------|
| a | 16 |
| b | 5 |
| c | 12 |
| d | 17 |
| e | 10 |
| f | 25 |

Użyj algorytmu Huffmana w celu skonstruowania optymalnego binarnego kodu prefiksowego dla powyższych liter.

Przykład 2

Założmy, że zbiór znaków ma postać {A, B, C, D, E} a ich prawdopodobieństwa wystąpienia przedstawia poniższa tabela:

| Litera | Prawdopodobieństwo |
|--------|--------------------|
| A | 0,39 |
| B | 0,21 |
| C | 0,19 |
| D | 0,12 |
| E | 0,09 |

Użyj algorytmu Huffmana w celu skonstruowania optymalnego binarnego kodu prefiksowego dla powyższych liter.

UWAGA

Nie można wykorzystać rozwiązań ze strony <http://eduinf.waw.pl/inf/> Zastosuj obsługę wyjątków.

Zaimplementuj własną klasę dla swoich wyjątków. Wykorzystaj klasę `runtime_error`.

ZASADY ODDAWANIA GOTOWYCH PROGRAMÓW:

Plik `.cpp` o nazwie: **Nazwisko_Imie_Program_01.cpp**

wraz z wszystkimi wyjściowymi plikami tekstowymi powinny być zamieszczone w katalogu: **Nazwisko_Imie_Laboratorium_10**

Katalog powinien być spakowany w formacie `.rar` lub `.zip` i przesłany do folderu: **Programy - laboratorium 10 – Poniedziałek godzina 16.15** dostępnego na stronie kursu MP (elf2.pk.edu.pl).

LITERATURA:

Neapolitan R. : Podstawy algorytmów z przykładami w C++, Helion 2004

Drozd A. : Algorytmy i struktury danych, Helion 2004