

Teoria informacji i kodowania - ćwiczenia laboratoryjne

Zadanie 2

Napisać program w języku C++ tworzący na podstawie modelu źródła informacji wyznaczonego wcześniej dla dowolnego pliku drzewo kodowania i tabelę kodową dla tego pliku.

W tym celu należy z pliku wejściowego wczytać do odpowiednio utworzonej tablicy tabelę symboli (bajtów 8-bitowych) posortowaną według liczby wystąpień i korzystając z metody Huffmana wyznaczyć i zapisać w pliku wynikowym tabelę opisującą drzewo kodowania, a następnie wyznaczyć na podstawie drzewa kodowania i zapisać w pliku wynikowym tabelę kodową.

Szczegółowe wymagania:

- program ma być uruchamiany z okna konsoli tekstowej, bez interfejsu graficznego;
- nazwa pliku wejściowego zawierającego model źródła informacji z rozszerzeniem ***.modelsort** ma być podawana jako parametr wejściowy przy uruchamianiu programu, plik wejściowy ma znajdować się w bieżącym katalogu roboczym;
- w wyniku działania programu w pliku wynikowym o nazwie takiej jak nazwa pliku wejściowego lub innej dowolnej, ale w każdym przypadku z rozszerzeniem ***.tree** ma być zapisane drzewo kodowania w postaci tabeli zawierającej poszczególne węzły drzewa (symbole podstawowe i zastępcze) wraz z informacją o ich połączeniach w formacie:

symbol - ojciec <spacja> symbol - potomek lewy <spacja> symbol - potomek prawy
symbol - ojciec <spacja> symbol - potomek lewy <spacja> symbol - potomek prawy
...
symbol - ojciec <spacja> symbol - potomek lewy <spacja> symbol - potomek prawy

Symbole podstawowe to liczby dodatnie w systemie dziesiętnym, a jako symbole zastępcze należy użyć kolejne liczby ujemne w systemie dziesiętnym od -1 w dół.

- w wyniku działania programu w pliku wynikowym o nazwie takiej jak nazwa pliku wejściowego lub innej dowolnej, ale w każdym przypadku z rozszerzeniem ***.coding** ma być zapisana tabela kodu zawierająca symbole podstawowe i zastępcze wraz z przyporządkowanymi im łańcuchami znaków kodujących (typ string) składających się z „0” i „1” (utworzone w porządku „0” dla lewego potomka, „1” dla prawego potomka) w formacie:

symbol podstawowy lub zastępczy <spacja> łańcuch znaków kodujących z „0” i „1”
symbol podstawowy lub zastępczy <spacja> łańcuch znaków kodujących z „0” i „1”
...
symbol podstawowy lub zastępczy <spacja> łańcuch znaków kodujących z „0” i „1”

- w wyniku działania programu w pliku wynikowym o nazwie takiej jak nazwa pliku wejściowego lub innej dowolnej, ale w każdym przypadku z rozszerzeniem ***.code** ma być zapisana tabela kodowa zawierająca tylko właściwe symbole podstawowe (bajty 8-bitowe) wraz z przyporządkowanymi im łańcuchami znaków kodujących (typ string) składających się z „0” i „1” w formacie:

symbol (dodatnia liczba w systemie dziesiętnym) <spacja> łańcuch znaków kodujących z „0” i „1”

symbol (dodatnia liczba w systemie dziesiętnym) <spacja> łańcuch znaków kodujących z „0” i „1”

...

symbol (dodatnia liczba w systemie dziesiętnym) <spacja> łańcuch znaków kodujących z „0” i „1”

Można - ale to nie jest wymagane - dla zwiększenia efektywności kodowania posortować tabelę kodową przed zapisaniem do pliku w takim porządku w jakim był na samym początku posortowany model źródła informacji.

- program ma informować na bieżąco użytkownika o wykonywanych operacjach w postaci informacji tekstowych wyświetlanych w oknie konsoli tekstowej;
- program ma rozpocząć i zakończyć swoje działanie bez potrzeby dodatkowych działań ze strony użytkownika (poza jego uruchomieniem);
- w programie poza funkcją **main()** mają być oddzielne funkcje z przekazywanymi do nich odpowiednimi parametrami co najmniej dla operacji wczytywania modelu źródła, wyznaczania drzewa kodowania oraz wyznaczania tabeli kodowej;
- do sortowania zastosować funkcje **qsort()** lub **sort()** albo samodzielnie zaimplementowane sortowanie bąbelkowe lub inne;
- w wersji **podstawowej** jako wszelkie tablice należy odpowiednio zastosować typ danych tablica statyczna;
- w wersji **zaawansowanej** jako wszelkie tablice należy zastosować listę dynamiczną zaimplementowaną samodzielnie lub skorzystać z gotowych kontenerów z biblioteki STL. W tej wersji można także zamiast odpowiednio utworzonej tablicy do przechowywania struktury drzewa kodowania zaimplementować strukturę danych pełnego drzewa binarnego korzystając ze wskaźników lub gotowych kontenerów z biblioteki STL.