Kodowanie Huffmana

Radomir Nowacki

Wydział Nauk Ekonomicznych

Uniwersytet Warszawski

18 maja 2024

Projekt zaliczeniowy

Zaawansowane Programowanie Komputerowe

dr Krzysztof Ziemiański

Opis projektu

Projekt dotyczy implementacji algorytmu kodowania Huffmana w języku C++. Kodowanie Huffmana to metoda kompresji danych, która polega na przypisaniu krótszego kodu binarnego częściej występującym znakom, a dłuższego tym, które występują rzadziej. Ta metoda kodowania została opracowana przez Davida Huffmana w 1952 roku podczas jego studiów doktoranckich na Massachusetts Institute of Technology.

Opis algorytmu

Kodowanie Huffmana zaimplementowane w programie odbywa się za pomocą poniższego algorytmu.

1. Dla wszystkich poszczególnych znaków w danym pliku wyliczana jest częstość ich wystepowania
2. Każda z par znak – częstość jest dodana do kolejki priorytetowej jako liść drzewa.
3. Usuń z kolejki dwa elementy o najmniejszych wartościach częstości i dodaj do kolejki węzeł z wartością częstości równą sumie usuniętych liści. Liście dołączane są do dodanego węzła
4. Krok 3 powtarzany jest tak długo jak liczba elementów w kolejce jest większa od jedności.
5. Na podstawie stworzonego drzewa Huffmana można utworzyć kod Huffmana. Poczynając od korzenia drzewa – każda lewa krawędź węzła oznacza dopisanie 0 do kodu, a prawa krawędź 1.

Instrukcja użytkowania

Program przed użyciem należy skompilować. Dla referencji poniżej znajduje się komenda uyżywana do kompilacji programu przez autora.

```
g++ -Wall -Wextra -Wpedantic -o huffman *.cpp
```

Po skompilowaniu wywołanie programu poinformuje użytkownika o poprawnym użyciu programu. W celu zakodowania teksut należy wywołać program z nastepującymi argumentami.

```
huffman -e <plik_wejściowy> -o <plik_wyjściowy>
```

Analogicznie w celu dekodowania.

```
huffman -d <plik_wejściowy> -o <plik_wyjściowy>
```

Opis kodu programu

Program składa się z pliku głównego huffman.cpp oraz klas HuffmanNode, HuffmanEncoder, HuffmanDecoder, BitWriter, BitReader, które zawierają odpowiednie funkcje do kodowania i dekodowania danych.

huffman.cpp

Plik główny programu który przetwarza argumenty wejściowe, wczytuje plik wejściowy, koduje go i zapisuje do pliku wyjściowego, kodując lub dekodując dane w zależności od argumentów wejściowych.

huffmanNode.h oraz huffmanNode.cpp

Pliki te zawierają klasę HuffmanNode, która reprezentuje węzeł drzewa Huffmana. Klasa zawiera pola ze znakiem i częstością występowania znaku, wskaźniki na lewe i prawe dziecko. Zawiera odpowiednie konstruktory, destruktory oraz klasę porównującą węzły.

huffmanEncoder.h oraz huffmanEncoder.cpp

Pliki te zawierają klasę HuffmanEncoder, która koduje dane wejściowe do postaci binarnej. W kolejnych krokach generowane są: mapa częstości występowania znaków, drzewo Huffmana, mapa kodów znaków. Następnie dane wejściowe są kodowane do postaci binarnej w następującej postaci: nagłówek zawierający informację o długości kodowanego tekstu, reprezentację drzewa Huffmana oraz zakodowane dane.

Reprezentacja drzewa Huffmana

Drzewo Huffmana jest reprezentowane w postaci binarnej w następujący sposób:

- 0 oznacza węzeł wewnętrzny
- 1 oznacza liść
- po 1 następuje 8 bitów reprezentujących znak

Reprezentacja ta pozwala na odtworzenie drzewa Huffmana potrzebnego do dekodowania danych. Nie jest to optymalny sposób reprezentacji drzewa, ale jest on prosty i wystarczający do zaimplementowania algorytmu.

huffmanDecoder.h oraz huffmanDecoder.cpp

Pliki te zawierają klasę HuffmanDecoder, która dekoduje dane zakodowane algorytmem Huffmana. W kolejnych krokach wykonywane są nastepujące czynności: odczytanie nagłówka zawierającego informację o długości kodowanego tekstu, odczytanie reprezentacji binarnej drzewa Huffmana, odtworzenie drzewa Huffmana i ostatecznie z jego pomocą dekodowanie danych.