TIMKoD – Lab 6 – Kompresja bezstratna – Metoda LZW

Opis pliku z zadaniami

Wszystkie zadania na zajęciach będą przekazywane w postaci plików `.pdf`, sformatowanych podobnie do tego dokumentu. Zadania będą różnego rodzaju. Za każdym razem będą one odpowiednio oznaczone:

- Zadania do wykonania na zajęciach oznaczone są symbolem $\triangle$ – nie są one punktowane, ale należy je wykonać w czasie zajęć.
- Punktowane zadania do wykonania na zajęciach oznaczone są symbolem $\diamond$ – należy je wykonać na zajęciach i zaprezentować prowadzącemu, w wypadku nie wykonania zadania w czasie zajęć lub nieobecności, zadanie staje się zadaniem do wykonania w domu ($\star$).
- Zadania do wykonania w domu oznaczone są symbolem $\star$ – są one punktowane, należy je dostarczyć w sposób podany przez prowadzącego i w wyznaczonym terminie (zwykle przed kolejnymi zajęciami).
- Zadania programistyczne można wykonywać w dowolnym języku programowania, używając jedynie biblioteki standardowej dostępnej dla tego języka.

Cel zajęć

Celem dzisiejszych zajęć jest implementacja popularnej metody kompresji LZW.

Przygotowanie do zajęć

- Do wykonania zadań potrzebne będą korpusy tekstowe oraz obraz, które można pobrać z katalogu dostępnego w odpowiedniej sekcji kursu.

1 LZW *10pt*◇

Treść

Metoda bezstratnej kompresji LZW (Lempel–Ziv–Welch) została wymyślona przez Terrego A. Welcha w 1982 roku. Jest ona usprawnieniem algorytmu LZ78 zaproponowanego przez Abrahama Lempela, Jacoba Ziva w 1978 roku. Metoda LZW jest bardzo łatwa do zaprogramowania i daje zazwyczaj bardzo dobre rezultaty. Choć nie daje gwarancji kompresji danych, w specyficznych wypadkach rozmiar danych wyjściowych może być większy niż wejściowych. Wykorzystywana jest m.in. w formacie zapisu grafiki GIF, w formacie PDF.

Używając szablonu programu do kompresji i dekompresji zaimplementuj metodę LZW.

- Przetestuj metodę na plikach "norm_wiki_sample.txt", "wiki_sample.txt" i "lena.bmp".
- Porównaj rozmiar powstałych plików dla przypadku bez ograniczenia na rozmiar słownika oraz z ograniczeniem rozmiaru słownika do 2^{12} i 2^{18} elementów.
- Implementacja powinna działać dla plików binarnych.

Źrodła

- https://en.wikipedia.org/wiki/Lempel%E2%80%93Ziv%E2%80%93Welch
- https://en.wikipedia.org/wiki/Huffman_coding
- http://www.inference.org.uk/itprnn/book.pdf