Run-Length & Huffman Encoding (Kodowanie bieżącej długości (ciągu) i kodowanie Huffmana)

Przemysław Śliwiński

May 19, 2017

1 Algorytm RLE

1. Zaimplementować algorytm kodowania (i dekodowania) ciągów liczb całkowitych **RLE** (ang. *run-length encoding*), który przekształca ciąg liczb w ciąg par liczb przechowujących wartości ciągu pierwotnego i liczby ich kolejnych powtórzeń,np.

$$\{1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,1,1,1,1,0,1,0,1\}$$
$$\overset{\text{RLE}}{\Longrightarrow}$$
$$\{\{1,11\},\{0,5\},\{1,4\},\{0,1\},\{1,1\},\{0,1\},\{1,1\}\}.$$
$$\overset{\text{Binary RLE}}{\Longrightarrow}$$
$$\{11,5,4,1,1,1,1\}.$$

2. Zastosować go do kompresji (i dekompresji) obrazów z katalogu:

   `https://1drv.ms/f/s!ArXn9WU5UgjzpNIzfV1fiLvWR602Yw`

3. Porównać, ocenić i wyjaśnić przyczyny efektywności (ew. jej braku) algorytmu na podstawie poszczególnych przykładów.

2 Algorytm Huffmana

1. Zakodować surowe obrazy z katalogu

   `https://1drv.ms/f/s!ArXn9WU5UgjzpNIzfV1fiLvWR602Yw`

   za pomoca alorytmu Huffmana.

2. Zakodować obrazy po zastosowaniu RLE.

3. Porównać wyniki i wyciągnąć wnioski.