

Proste algorytmy kompresji bezstratnej. **Huffman vs RLE + Huffman**

dr hab. inż. Przemysław Śliwiński

December 4, 2015

1 Algorytm RLE

1. Zaimplementować algorytm kompresji wybranych 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.

$$\{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 1, 1, 1, 1, 0, 1, 0, 1\}$$
$$\xrightarrow{\text{RLE}}$$
$$\{\{1, 11\}, \{2, 5\}, \{1, 4\}, \{0, 1\}, \{1, 1\}, \{0, 1\}, \{1, 1\}\}.$$

2. Pomierzyć efektywność algorytmu wyrażoną przez stosunek długości ciągu wejściowego do wyjściowego:
 - (a) zakodować wynik transformacji i kwantyzacji z poprzedniego laboratorium za pomocą kodu Huffmana (np. za pomocą funkcji `'huffmandeco'` w Matlabie)
 - (b) uzyskany ciąg zer i jedynek zakodować za pomocą RLE i ponownie zakodować za pomocą kodu Huffmana.
 - (c) sformułować wnioski.