

Abstract

Yasser El Sonbaty

Efficient Algorithms for Updating Huffman Codes

given a list $w = [w_1, \dots, w_n]$ of n positive integer symbol weights, a list $l = [l_1, \dots, l_n]$ of n corresponding integer codeword lengths, it is required to find the new list l when a new value x is added when an existing value is d from the list of weights w . the presented algorithm uses the given information about the weights and their corresponding levels in order to perform the required update. no other knowledge about the original Huffman tree is assumed to be known. instead of rebuilding the Huffman tree, the new algorithm redistributes the weights among the levels to obtain the new Huffman code. in many special cases, the new Huffman code can be generated with lower complexity than reconstructing the Huffman tree from scratch by efficiently using the information of weights and their levels. in this paper, we present an updating algorithm that requires a linear complexity in many practical cases rather than the $O(n \log n)$ needed for reconstructing the Huffman tree. we also give a practical $O(n \log n)$ implementation for our algorithms.