

Структура дерева кодирования

Теория

Дерево кодирования представляет собой бинарное дерево Хаффмана. Оно строится на основе частоты повторения символов в исходном тексте.

Дерево состоит из листьев (конечных узлов) и внутренних узлов. Каждый лист дерева представляет уникальный символ из исходного текста, а внутренние узлы содержат суммарные частоты.

Кодирование символов осуществляется путем обхода дерева от корня к листьям, где следование по левому направлению обозначено «0», по правому – «1».

Основная идея построения дерева заключается в том, чтобы для наиболее часто встречающихся символов их кодовая строка имела меньше битовых символов, чем у символов встречающихся реже.

Структура класса Node

Класс Node представляет узел дерева кодирования и содержит следующие поля: `character` – символ (если узел является листом); `frequency` – частота символа в тексте; `left` – ссылка на левого потомка; `right` – ссылка на правого потомка.

Построение дерева Хаффмана

Дерево строится в методе `build_Huffman_tree` класса `huffman_encoder`. Он использует приоритетную очередь (`PriorityQueue`), где узлы сравниваются по их частотам. Построение происходит в несколько этапов: сначала создаются узлы для каждого символа, затем они объединяются в дерево по следующему принципу: из узлов в приоритетной очереди выбираются два с наименьшей частотой, выбранные узлы удаляются из очереди, вместо них создается узел имеющий в своих дочерних узлах эти удаленные узлы и с частотой равной суммарной частоте удаленных узлов, этот созданный узел кладется в очередь. Это объединение производится до тех пор, пока не останется один корневой узел.

Структура дерева кодирования

Структура дерева кодирования представляет собой набор строк, где каждая строка содержит информацию об одном символе и его коде Хаффмана. Каждая строка выводится в формате: «символ код». Создание данного дерева кодирования производится в классе `huffman_encode` в рекурсивной функции `coding`.