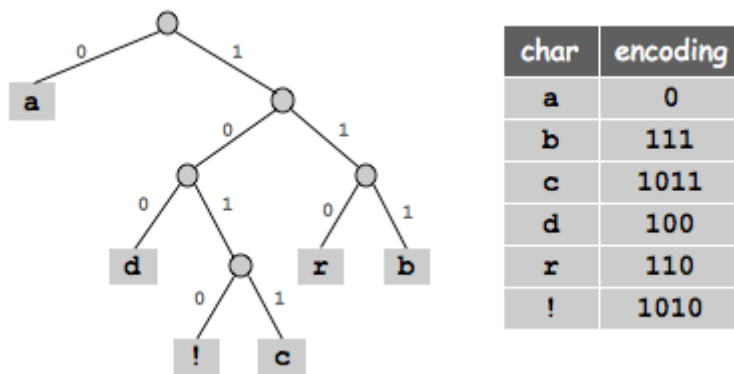


Описание структуры дерева кодирования

Общая информация:

В данной работе используется алгоритм кодирования Хаффмана - это жадный алгоритм оптимального префиксного кодирования алфавита (ни одно кодовое слово не является началом другого). В листах дерева хранятся символы, определив правило (например, при движении налево будем добавлять 0, а при движении направо 1) можем составить однозначный битовый код для каждого из данных символов. Узлы дерева являются пустыми и содержат только ссылки на своих потомков.



Пример дерева Хаффмана

Структура элемент дерева:

Класс элемента дерева содержит следующие поля:

- Символ (для узлов равняется нулю)
- Строковую запись битового кода символа (для узлов пустая)
- Левый и правый потомки (для листов равны нулю)

Правило построения дерева:

Для построения дерева необходимо составить ассоциативный список символов с их весом (частота появления в тексте). В данной реализации используется хэш-таблица. Требуется минимизировать число битов для кодирования символов с наибольшим весом.

Стратегия построения – заполним контейнер типа мультимэпы, поддерживающий возможность получение элемента с наименьшим ключом. В качестве ключей будут выступать веса символов, а значениями будут элементами дерева. Будем забирать из мультимэпы элементы с наименьшими ключами, добавлять их в качестве потомков новой вершины, после чего поместим её обратно в мультимэпу (её вес будет равен сумме весов потомков).

Для достижения лучшей алгоритмической сложности используются стандартные средства JDK, реализующие красно-черное дерево и односвязный список.

Представление дерева:

- В виде словаря (хэш-таблица: символ строка)
- С помощью класса HuffmanTree