

Метод Шеннона-Фано

Группа М40-209Б-18

Савин Н.А

Лыков М.Н

Идея метода Шеннона-Фано

- Идея этого метода заключалась в том, чтобы заменить часто встречающиеся символы более короткими кодами, а редко встречающиеся последовательности более длинными кодами. Этот алгоритм основывается на кодах переменной длины. Для того чтобы впоследствии можно было без проблем раскодировать сжатую последовательность каждый код должен обладать уникальностью, а именно, не смотря на их переменную длину, код должен уникально определять один закодированный символ и не являться префиксом любого другого кода.
- Метод Шеннона-Фано относится к вероятностным методам сжатия

Сравнительная характеристика метода Шеннона-Фано и метода Хаффмана

- Кодирование по методу Шеннона-Фано не всегда приводит к однозначному построению кода, так как при разбиении на подгруппы на первой итерации можно сделать большей по вероятности как верхнюю, так и нижнюю подгруппу. В результате чего среднее число символов на букву окажется другим. Из-за этого, построенный код может оказаться не самым лучшим.
- В свою очередь кодирование методом Хаффмана гарантирует однозначное построение кода с наименьшим для данного распределения вероятностей средним числом символов на букву. Метод Хаффмана производит идеальное сжатие (сжимает данные до их энтропии), если вероятности символов точно равны отрицательным степеням числа 2.

Плюсы и минусы метода Шеннона-Фано

- Плюсы:
 - 1. не нужен символ разделитель.
 - 2. Учитывается частота символов
 - 3. Код префиксный – можно декодировать по мере поступления данных
- Минусы:
 - 1. необходима таблица частот
 - 2. не учитывает повторяющиеся последовательности символов
 - 3. код не оптимален поскольку требует использования дерева
 - 4. при ошибке в передаче сложно восстановить «хвост»

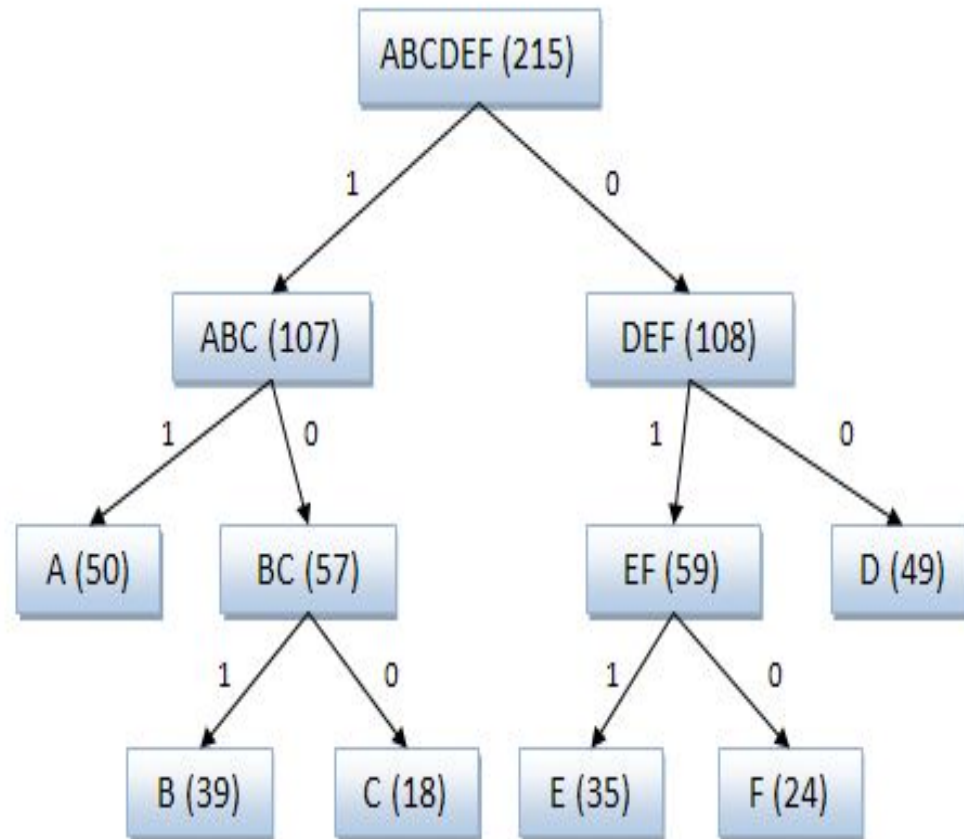
Основные этапы

- Все встречающиеся символы в данном файле записываются в порядке убывания частоты их появления.
- Символы полученного алфавита делят на две части, суммарные вероятности символов которых максимально близки друг другу
- В префиксном коде для первой части алфавита присваивается двоичная цифра «0», для второй части – «1».
- Полученные части рекурсивно делятся и их частям назначаются соответствующие двоичные цифры в префиксном коде.
- Когда размер подалфавита становится равен нулю или единице, то дальнейшего удлинения префиксного кода для соответствующих ему символов первичного алфавита не происходит, таким образом, алгоритм присваивает различным символам префиксные коды разной длины. На шаге деления алфавита существует неоднозначность, так как разность суммарных вероятностей p_0, p_1 может быть одинакова для двух вариантов разделения (учитывая, что все символы первичного алфавита имеют вероятность больше нуля).

a_i	$p(a_i)$	1	2	3	4	Итого	
a_1	0.36	0	00			00	
a_2	0.18		01			01	
a_3	0.18	1	10			10	
a_4	0.12		110			110	
a_5	0.09		11	111	1110		1110
a_6	0.07				1111		1111

Пример кодового дерева

- A – частота 50
- B – частота 39
- C – частота 18
- D – частота 49
- E – частота 35
- F – частота 24



Получился код:

A - 11

B - 101

C - 100

D - 00

E - 011

F - 010