Конспект

Сегодня говорим про неравномерное кодирование.

Принципиальное отличие от равномерного кодирования одно – при равномерном кодировании каждое значение получает код одинаковой длины, при неравномерном кодировании длина может быть разной, то есть какой-то код может быть короче, какой-то длиннее.

Разберемся зачем это нужно на примере английского алфавита и частотности встречаемости букв в нем. Буквы e, a ,t, o встречаются очень часто, z, q, x – которые встречаются очень редко. И разница между этими частотностями очень большая. Чтобы сэкономить (сжать информацию) пространство для хранения мы можем буковки, которые встречаются часто, закодировать коротким кодом, буквы, которые встречаются редко, придется закодировать кодом длиннее. И, за счет того, что буквы с короткими кодами встречаются часто, а с длинными – редко, мы на самом деле сэкономим. То есть итоговый размер текстового документа будет меньше. В реальном хранении такой способ, конечно, не используется, но такой способ может быть применен, например, при архивации.

Преимущество равномерного кодирования – легкие чтение и запись. Неравномерное кодирование же не дает такой простоты. Например, азбука Морзе. Где Л – точка, черточка и две точки, то такой код мы можем расшифровать несколькими способами – АИ (.- и ..), Л (.-..), ЕТЕЕ(., -, ., .). И это проблема, которую необходимо решать при использовании неравномерного кодирования.

Проблема однозначного кодирования

А •–	Л •–••	Ц –•–•
Б –•••	М ––	Ч –––•
В •––	Н –•	Ш ––––
Г ––•	О –––	Щ ––•–
Д –••	П •––•	Ъ ––•––•
Е •	Р •–•	Ы –•––
Ж •••–	С •••	Ь –••–
З ––••	Т –	Э ••–••
И ••	У ••–	Ю ••––
Й •–––	Ф ••–•	Я •–•–
К –•–	Х ••••	

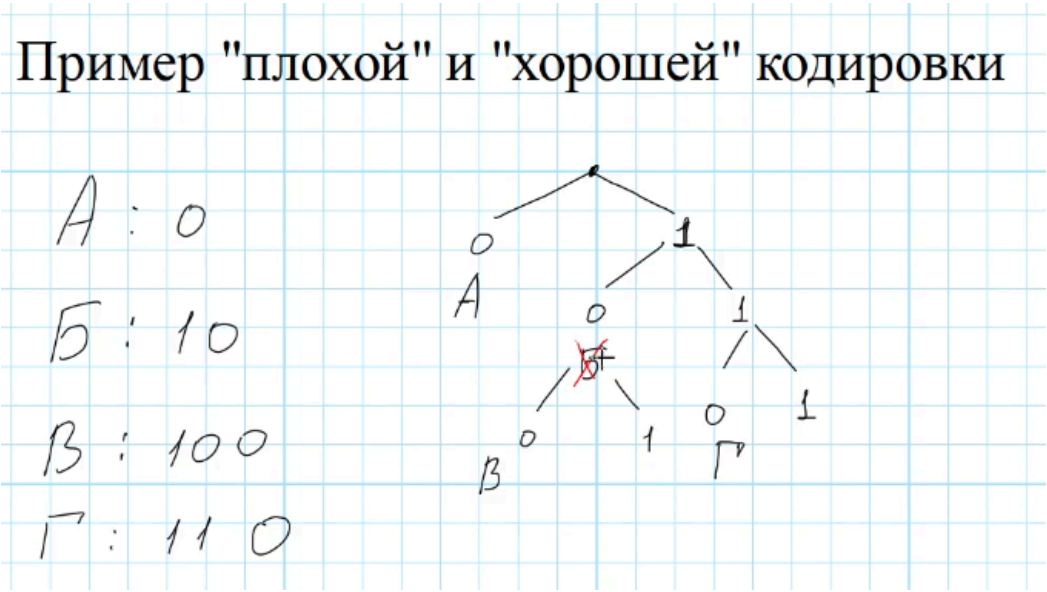
Основное условие однозначного декодирования, которым мы будем пользоваться, - условие Фано: «Любое сообщение будет однозначно декодировано, если никакое кодовое слово не является началом другого кодового слова». Если кодировка удовлетворяет такому условию, то любое сообщение гарантировано может быть прочитано единственным образом.

Начнем с примеров «плохой» и «хорошей» кодировок. А: 0, Б: 10, В: 100, Г: 110

Почему кодировка плохая? Код для буквы Б является началом для кода буквы В. Поэтому такую кодировку использовать нельзя.

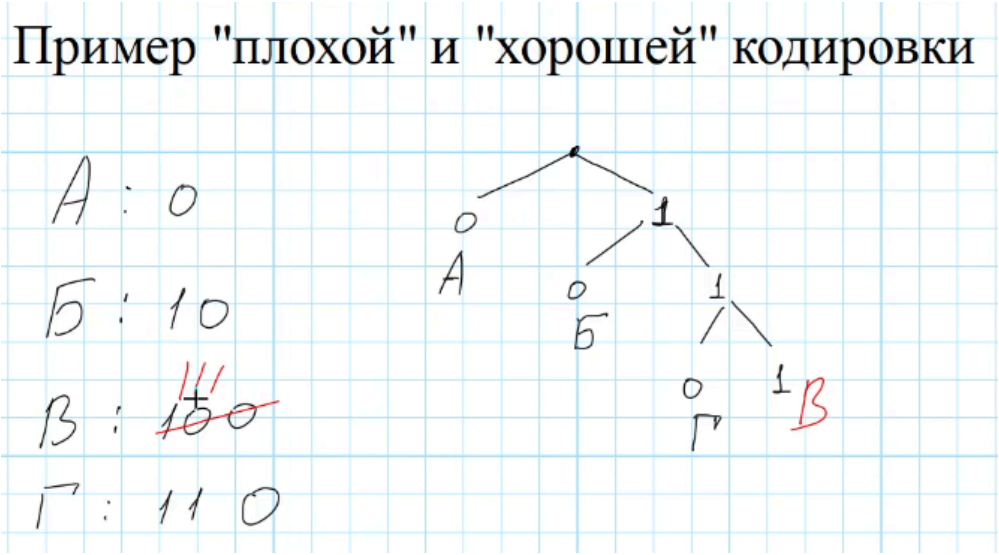
Попробуем исправить кодировку, использовав двоичное дерево.

Дерево рисуется от корня. От него делается два ответвления – 0 и 1. А – это 0, обозначим эту ветку. На 0 другие коды не начинаются. От ветки 1 сделаем две ветви 0 и 1. 10 – это Б, обозначим это на дереве. Разветвим коды 10 и 11 и расставим оставшиеся буквы.



Сразу бросается в глаза, что буква Б является началом для буквы В. Поэтому нужно переставить либо букву Б или букву В. Можно поставить Б в место 111. Тогда будет соблюдаться условие Фано.

Но мы можем и переместить код для буквы В в 111. Тогда и буква Б будет иметь код не длиннее, чем изначально, и В будет удовлетворять условию Фано.



Исследуем полученное дерево. Можно заметить, что свободных ответвлений не осталось. И, если я захочу добавить букву Д, то я не смогу её закодировать, так как новую букву некуда добавить.

Чтобы кодовую таблицу можно было дополнять, мы можем сделать развилку в одной из букв и добавить свободное место (100). Так у нас появилась возможность закодировать любое количество новых букв.

Пример "плохой" и "хорошей" кодировки

А : 0

Б : 10

В : ~~100~~ 111

Г : 110

Д :