### **Отчет**

Репозиторий: <https://github.com/F4nnur/semestr-work-TIK-Akhmetov>



1. Алгоритм Шеннона

В области сжатия данных, код Шеннона, названный в честь его создателя, Клода Шеннона, — это алгоритм сжатия данных без потерь с помощью построения префиксных кодов на основе набора символов и их вероятностей (расчётное или измеренное). Он является субоптимальным в том смысле, что не позволяет достичь минимально возможных кодовых длин как в кодировании Хаффмана, и никогда не будет лучше, но иногда равным с кодом Шеннона-Фано.

В кодировании Шеннона символы располагаются в порядке от наиболее вероятных к наименее вероятным. Им присваиваются коды, путем взятия первых l_i = \left\lceil {-\log}  p_i \right\rceil  цифр из двоичного разложения кумулятивной вероятности  \sum\limits_{k=1}^{i-1} p_k . Здесь \left\lceil x \right\rceil обозначает функцию, которая округляет x до ближайшего целого значения.

1. Алгоритм декодирования БУ

Преобразование Барроуза — Уилера (англ. Burrows-Wheeler transform) — алгоритм, используемый для предварительной обработки данных перед сжатием, разработанный для улучшения эффективности последующего кодирования. Преобразование Барроуза — Уилера меняет порядок символов во входной строке таким образом, что повторяющиеся подстроки образуют на выходе идущие подряд последовательности одинаковых символов.

Пусть нам дано: BWT(s)=("BCABAAA", 3). Тогда выпишем в столбик нашу преобразованную последовательность символов "BCABAAA". Запишем её как последний столбик предыдущей матрицы (при прямом преобразовании Барроуза — Уилера), при этом все предыдущие столбцы оставляем пустыми. Далее построчно отсортируем матрицу, затем в предыдущий столбец запишем "BCABAAA". Опять построчно отсортируем матрицу. Продолжая таким образом, можно восстановить полный список всех циклических сдвигов строки, которую нам надо найти. Выстроив полный отсортированный список сдвигов, выберем строку с номером, который нам был изначально дан. В итоге мы получим искомую строку. Алгоритм обратного преобразования описан в таблице ниже:

3.Стопка книг

Основной идеей преобразования является замена каждого входного символа его номером в специальном стеке недавно использованных символов. Последовательности идентичных символов, к примеру, будут заменены (начиная со второго символа) на последовательность нулей. Если же символ долго не появлялся во входной последовательности, он будет заменён большим числом. Преобразование заменяет последовательность входных символов на последовательность целых чисел, если во входных данных было много локальных корреляций, то среди этих чисел будут преобладать небольшие, лучше сжимаемые энтропийным кодированием, чем исходные данные.

Изначально алгоритм назывался «стопка книг» («book stack»). История разработки алгоритма рассказана в [2].

Часто используется при преобразовании байтов. Изначально каждое возможное значение байта записывается в список, в ячейку с номером, равным значению байта, т.е. (0, 1, 2, 3, …, 255). В процессе обработки данных этот список изменяется. Первый обработанный символ заменяется самим собой, после чего элемент, соответствующий этому символу, перемещается в голову списка (сдвигая элементы с 0 по своё положение на 1 вправо). Последующие символы кодируются номером элемента, содержащего их значение. После кодирования каждого символа эти элементы также продвигаются к голове списка.