Коды и сжатие данных. Задачи. Вариант с подсказками.

Предисловие

При расшифровке мы читаем закодированную последовательность 0 и 1 и решаем, из каких букв исходного алфавита она получилась. Если код буквы "a" совпадает с началом кода буквы "b", это затрудняет расшифровку.

Удобно, когда, прочтя код буквы, мы сразу можем ее распознать.

Такие коды называют префиксными -- от prefix, английского слова "приставка" .

Префиксный код гарантирует кодирование без потерь: из разных текстов получатся разные коды.

Пример: код { 0, 10, 11 } является префиксным.

Поэтому сообщение 01001101110 можно разбить на слова единственным образом: `0 10 0 11 0 11 10`.

Упражнения

1. Однозначна ли расшифровка кода из 0 и 1 ?

а) C_1 = { 0, 101 }

ПОДСКАЗКА: 0 не является началом 101. Код префиксный.

б) C_2 = { 1, 101 }

ПОДСКАЗКА: 1 является началом 101. Код НЕ префиксный. Кодирует без потерь.

в) C_3 = { 0, 10, 110, 111 }

ПОДСКАЗКА: - префиксный.

г) C_4 = { 00, 01, 10, 11 }

ПОДСКАЗКА: - префиксный.

д) C_5 = { 00, 11, 0101, 111, 1010, 100100, 0110 } ?

Контрпример: 11111 --> 111 11 и 11 111;

е) Код из 0, 1 и 2: { 00, 012, 0110, 0112, 100, 201, 212, 22 } ?

ПОДСКАЗКА: - префиксный.

2. Подадим на вход коду без потерь всевозможные комбинации из N символов.
Например, при N=4: "0000", "0001", "0010", "0011", ... , "1111".
Предложите код, который каждую исходную последовательность отображает в более короткую.
Например, "0000" --> "0", "0001" --> "1", ...

НЕВОЗМОЖНО по принципу Дирихле.

3. Если известен набор длин кодовых слов, то по ним можно судить о возможности кодировать без потерь.
Например, можно вычислить сумму, получающуюся из длин кодовых слов $S = \sum_{i \in C} 2^{-l_i}$.

Найдите ее для задач 1 а,в,г и для кода { 0, 10, 11, 100 }.
Что можно сказать о коде, если сумма больше 1?
Обоснуйте или докажите.

ЕСЛИ сумма > 1, то нет префиксности и нет взаимной однозначности.
В классе были интуитивные соображения. Руслан Фаддеев сопоставил дерево
каждому префиксному коду и сказал, что для префиксности нужно,
чтобы код содержал только листья дерева. В этом случае сумма не превосходит 1.
Значит, если сумма > 1, то префиксноости нет.

4. Пусть алфавит состоит из букв "0" и "1", A = {0, 1}, их вероятности равны P = {0.9, 0.1}.
Сделайте код Хаффмана из 0 и 1, для
а) всех слов длины 2: {00, 01, 10, 11};
б) всех слов длины 3: {000, 001, 010, 011, 100, 101, 110, 111}.
* Для каждого кода найдите среднюю ожидаемую длину закодированного слова.

* Какой код обеспечивает лучшее сжатие:
(средняя длина закодированного слова **)** **/ (** длина незакодированного слова: 2 для а) и 3 для б) **)**

Большая длина слова позволяет достичь большей степени сжатия.
ОТНОШЕНИЕ ДЛИН ЧУТЬ БОЛЬШЕ 0.6 В СЛУЧАЕ а)
ОТНОШЕНИЕ ДЛИН ЧУТЬ МЕНЬШЕ 0.6 В СЛУЧАЕ б).

5. Предложите код Хаффмана, который кодирует сообщение не двумя символами 0 и 1, а тремя: 0, 1 и 2.
Закодируйте один из примеров предыдущей задачи новым кодом.
Является ли лёгким для расшифровки код из 0, 1 и 2: { 00, 012, 0110, 0112, 100, 201, 212, 22 } ?

Многие сделали код для {00, 01, 01, 11} --> {0, 10, 11, 12} и заметили, что он не оптимален:
его можно преобразовать в более кроткий: {0, 1, 21, 22}.
При обсуждении, предположили, что проблема это из-за того, что 4 не делится на 3.
Оптимальная процедура осталась неизвестной.

6. Для алфавита из 4 символов {1, 2, 3, 4}, приведите пример 4 вероятностей,
{ p_1, p_2, p_3, p_4 } -- таких, что можно построить два кода Хаффмана,
имеющие разные наборы длин { L_1, L_2, L_3, L_4 }.

Задача оказалось трудной: было непонимание условия.
Несколько человек разобрались с подсказкой.
ПРИМЕР: набор вероятностей
{1/3, 1/3, 1/6, 1/6} после первого шага приводит к набору
{1/3, 1/3, 1/3} и назначенным символам: { , , (0, 1) }.
Второй шаг можно сделать, объединив пары символов разными способами.
В зависимости от способа объединения, получаются разные коды с разными наборами длин символов.
{ 00, 01, 10, 11 } и
{ 1, 01, 001, 000 }.

ЕСТЬ ЕЩЁ примеры наборов вероятностей: { 1/5, 1/5, 1/5, 2/5 } и {1/3, 1/3, 1/3, 0}.
Можно вынести в отдельные задачи.