

Сжатие

Определение

Сжатие – процесс сокращения битов (символов), необходимых для хранения одной и той же информации. **Информация не меняется**, но меняется количество бит.

Код Шенона-Фано

Алгоритм, основанный на деревьях.

1. Построение таблицы частот символов (A – 50, B – 39, C – 18, D – 49, E – 35, F – 24).
2. Построение дерева:
 - корень == все символы;
 - делим частоты по два

```
ABC ----- DEF
A -- BC      EF -- D
B -- C      E -- F
```

3. Присваиваем коды, путь направо – 0, налево – 1 (A – 11, B – 101, C – 100, D – 0).
4. Кодирование – буква → код.
5. Раскодирование – из корня до листа (получен символ), и снова из корня.

Алгоритм Хаффмана

Отличается второй этап предыдущего алгоритма: взять два символа с минимальной частотой → узел ($C + F \rightarrow (18 + 24) = 42$). Строит дерево не сверху, а от листьев к корню.

Недостатки

1. Требуется для шифрования два прохода (то есть сначала нужно построить таблицу частот, и только после этого можно приступить к сжатию).

2. Для расшифровки нужно дерево (для расшифровки можно использовать либо таблицу частот, либо дерево).

Адаптивное сжатие Хаффмана

Что предложено:

- инициализация дерева всеми символами;
- добавляемый символ отмечается в сжатом файле (при обновлении дерева);

Решены обе проблемы, в остальном это тот же алгоритм Хаффмана.

Лабораторная работа 7

Реализовать алгоритм Хаффмана (таблицу частот хранить в том же файле).

Алгоритм LZW

Последовательность действий:

1. Строим словарь всех символов (256) (0, 1, 2, ..., 255).
2. $W = M(0)$
3. $K = M(i)$
4. Находим в словаре $W + K$. Если не нашлось, то выводим в сжатый файл W , иначе – $(W + K)$ в словарь.

Например, docdocdoc

256	d	o
257	o	c
258	c	d
259	256	c
260	c	d

$M = \text{docdocdoc}$

$C = \text{doc } 256 \text{ с } 259$

5. Вывод второго столбца в сжатый файл.

Лабораторная работа 8

Реализовать алгоритм LZW

Арифметическое сжатие

1. Вероятность "встретить символ" (A – 60%, B – 20%, C – 10%, D – 10%).
2. Назначаются интервалы на участке от 0 до 1 (A – [0 - 0.6), B – [0.6 - 0.8), C – [0.8 - 0.9), D – [0.9 - 1]).
3. Для назначенных интервалов производим шифрование (C в интервале L и R):
 - $L = 0, R = 1, I = R - L;$
 - $L = L + I * L(i), R = L + I * R(i).$

Например, ACD

$L = 0, R = 1$

Считываем A

$L = 0 + 1 * 0 = 0$

$R = 0 + 1 * 0.6$

Считываем C

$L = 0 + 0.6 * 0.8 = 0.48$

$R = 0 + 0.6 * 0.9 = 0.54$

Считываем D

$L = 0.48 + 0.06 * 0.9 = 0.534$

$R = 0.48 + 0.06 * 1 = 0.54$

4. Разархивация C:

- $I = R - L;$
- $C = (C - L(i)) / I$