

Сжатие данных

ПОДГОТОВИЛ МАТВЕЕВ РОМАН

Сжатие с потерями и без потерь

Сжатие с потерями чаще используется, когда данные в результате оцениваются как нечто целое

Применять сжатие без потерь следует когда необходимо точно воспроизвести изначальный набор данных

С потерями

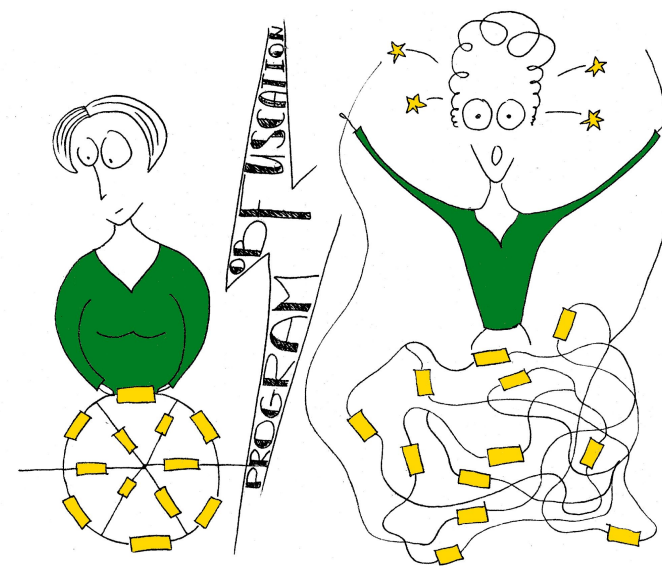


Без потерь



Сжатие пустых мест

- Очень часто после удаления пустых мест из передаваемой информации пропадает удобство чтения полученного сообщения. Дело в том, что исчезает та самая избыточность информации.



Групповое кодирование

Данный метод подразумевает использование итераторов для указания числа повторений символов в информационном сообщении

```
1 | "Y" Iter(8) "X"  
2 | 01001111 10001000 01011000
```

Кодирование по методу Хаффмана

- ▶ В этом методе для кодирования наиболее частых символов используется меньшее количество бит и наоборот: для наименее вероятных символов - большее кол-во бит

	Encoding	Symbol
1		
2	1	7
3	010	2
4	011	3
5	00000	4
6	00001	5
7	00010	6
8	00011	8
9	00100	9
10	00101	0
11	00111	1

Сжатие по алгоритму Лемпеля-Зива

Текущая строка	Текущий символ	Следующий символ	Вывод		Словарь
			Код	Биты	
ab	a	b	0	000	5: ab
ba	b	a	1	001	6: ba
ac	a	c	0	000	7: ac
ca	c	a	2	010	8: ca
ab	a	b	-	-	- -
aba	b	a	5	101	9: aba
ad	a	d	0	000	10: ad
da	d	a	3	011	11: da
ab	a	b	-	-	- -
aba	b	a	-	-	- -
abac	a	c	9	1001	12: abac
ca	c	a	-	-	- -
cab	a	b	8	1000	13: cab
ba	b	a	-	-	- -
bae	a	e	6	0110	14: bae
e	e	-	4	0100	- -

Данные		На выходе	Новая запись	
Биты	Код		Полная	Частичная
000	0	a	- -	5: a?
001	1	b	5: ab	6: b?
000	0	a	6: ba	7: a?
010	2	c	7: ac	8: c?
101	5	ab	8: ca	9: ab?
000	0	a	9: aba	10: a?
011	3	d	10: ad	11: d?
1001	9	aba	11: da	12: aba?
1000	8	ca	12: abac	13: ca?
0110	6	ba	13: cab	14: ba?
0100	4	e	14: bae	- -

- Идея, заложенная в алгоритме LZ78, заключается в кодировании потоковой последовательности байтов с использованием некоторой динамической таблицы.

Спасибо за внимание