

7. Словарные коды класса LZ

- **Алгоритмы словарного сжатия Зива-Лемпела появились во второй половине 1970-х гг.**

- **Словарные коды класса LZ широко используются в практических задачах.**
- **На их основе реализовано множество программ-архиваторов.**
- **Эти методы также используются при сжатии изображений в модемах и других цифровых устройствах передачи и хранения информации.**

- Словарные методы сжатия данных позволяют эффективно кодировать источники с неизвестной или меняющейся статистикой.
- Важными свойствами этих методов являются высокая скорость кодирования и декодирования, а также относительно небольшая сложность реализации.
- Кроме того, LZ-методы обладают способностью быстро адаптироваться к изменению статистической структуры сообщений.

- Словарные алгоритмы Зива - Лемпела разделяют на два семейства –
 - алгоритмы типа LZ77
 - алгоритмы типа LZ78.
- Иногда так-же говорят о словарных методах LZ1 и LZ2

Алгоритм LZ77

- Этот словарный алгоритм сжатия является самым старым среди методов LZ. Описание было опубликовано в 1977 г., но сам алгоритм разработан не позднее 1975 г.
- Алгоритм LZ77 является родоначальником целого семейства словарных схем - так называемых *алгоритмов со скользящим словарем*, или *скользящим окном*.

- В LZ77 в качестве словаря используется блок уже закодированной последовательности.
- Как правило, по мере выполнения обработки положение этого блока относительно начала последовательности постоянно меняется, словарь "скользит" по входному потоку данных.

- Скользящее окно имеет длину N , т. е. в него помещается N символов, и состоит из двух частей:
 - последовательности длины $W=N-n$ уже закодированных символов, которая и является словарем;
 - буфера предварительного просмотра длины n ; обычно n на порядки меньше W

- Алгоритм пытается найти в словаре фрагмент, совпадающий с содержимым буфера.
- Алгоритм выдает коды, состоящие из трех элементов
 - смещение в словаре относительно его начала подстроки, совпадающей с началом содержимого буфера
 - длина этой подстроки
 - первый символ буфера, следующий за подстрокой

- Затем *окно* смещается на длину закодированной подстроки символов вправо и осуществляется переход к новому циклу кодирования.

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

[illegible]

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

[illegible]

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

[illegible]

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

[illegible]

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

[illegible]

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

[illegible]

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

[illegible]

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

[illegible]

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

[illegible]

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

[illegible]

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

Словарь (16 символов)																Буфер (7 сим.)							Код
																к	о	л	о	к	о	л	<0,0,'к'>
														к		о	л	о	к	о	л	_	<0,0,'о'>
													к	о		л	о	к	о	л	_	о	<0,0,'л'>
												к	о	л		о	к	о	л	_	о	к	<1,1,'к'>
											к	о	л	о	к	о	л	_	о	к	о	л	<1,2,'_'>
								к	о	л	о	к	о	л	_	о	к	о	л	о	_	к	<3,4,'о'>
			к	о	л	о	к	о	л	_	о	к	о	л	о	_	к	о	л	о	к	о	<7,1,'к'>
	к	о	л	о	к	о	л	_	о	к	о	л	о	_	к	о	л	о	к	о	л	ь	<1,6,'ь'>
л	_	о	к	о	л	о	_	к	о	л	о	к	о	л	ь	н	и	:)				<0,0,'н'>
_	о	к	о	л	о	_	к	о	л	о	к	о	л	ь	н	и	:)					<0,0,'и'>
о	к	о	л	о	_	к	о	л	о	к	о	л	ь	н	и	:)						<0,0,': '>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

Словарь (16 символов)																Буфер (7 сим.)							Код
																к	о	л	о	к	о	л	<0,0,'к'>
														к		о	л	о	к	о	л	_	<0,0,'о'>
													к	о		л	о	к	о	л	_	о	<0,0,'л'>
												к	о	л		о	к	о	л	_	о	к	<1,1,'к'>
											к	о	л	о	к	о	л	_	о	к	о	л	<1,2,'_'>
								к	о	л	о	к	о	л	_	о	к	о	л	о	_	к	<3,4,'о'>
			к	о	л	о	к	о	л	_	о	к	о	л	о	_	к	о	л	о	к	о	<7,1,'к'>
	к	о	л	о	к	о	л	_	о	к	о	л	о	_	к	о	л	о	к	о	л	ь	<1,6,'ь'>
л	_	о	к	о	л	о	_	к	о	л	о	к	о	л	ь	н	и	:)				<0,0,'н'>
_	о	к	о	л	о	_	к	о	л	о	к	о	л	ь	н	и	:)					<0,0,'и'>
о	к	о	л	о	_	к	о	л	о	к	о	л	ь	н	и	:)						<0,0,': '>
к	о	л	о	_	к	о	л	о	к	о	л	ь	н	и	:)							<0,0,') '>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

Словарь (16 символов)															Буфер (7 сим.)							Код	
															к	о	л	о	к	о	л	<0,0,'к'>	
														к	о	л	о	к	о	л	_	<0,0,'о'>	
													к	о	л	о	к	о	л	_	о	<0,0,'л'>	
												к	о	л	о	к	о	л	_	о	к	<1,1,'к'>	
											к	о	л	о	к	о	л	_	о	к	о	л	<1,2,'_'>
								к	о	л	о	к	о	л	_	о	к	о	л	о	_	к	<3,4,'о'>
			к	о	л	о	к	о	л	_	о	к	о	л	о	_	к	о	л	о	к	о	<7,1,'к'>
	к	о	л	о	к	о	л	_	о	к	о	л	о	_	к	о	л	о	к	о	л	ь	<1,6,'ь'>
л	_	о	к	о	л	о	_	к	о	л	о	к	о	л	ь	н	и	:)				<0,0,'н'>
_	о	к	о	л	о	_	к	о	л	о	к	о	л	ь	н	и	:)					<0,0,'и'>
о	к	о	л	о	_	к	о	л	о	к	о	л	ь	н	и	:)						<0,0,':'>
к	о	л	о	_	к	о	л	о	к	о	л	ь	н	и	:)							<0,0,')'>
о	л	о	_	к	о	л	о	к	о	л	ь	н	и	:)								

- Длина кодов, определяется размером словаря и размером буфера

Алгоритм LZSS

- Алгоритм LZSS позволяет достаточно гибко сочетать в выходной последовательности символы и указатели (коды фраз), что до некоторой степени устраняет присущую LZ77 расточительность, проявляющуюся в регулярной передаче одного символа в прямом виде.
- Эта модификация LZ77 была предложена в 1982 г. Сторером (Storer) и Жимански (Szymanski)

- Идея алгоритма заключается в добавление к каждому указателю и символу 1-битового префикса, позволяющего различать эти объекты. Иначе говоря, 1-битовый флаг указывает тип и длину непосредственно следующих за ним данных.

Такая техника позволяет:

- записывать символы в явном виде, когда соответствующий им код имеет большую длину и, следовательно, словарное кодирование только вредит;
- обрабатывать ни разу не встреченные до текущего момента символы

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

Словарь (16 символов)																Буфер							Код LZSS				
																к	о	л	о	к	о	л	0,'к'				
															к	о	л	о	к	о	л	_		0,'о'			
														к	о	л	о	к	о	л	_		о	0,'л'			
													к	о	л	о	к	о	л	_		о	к	1,<1,1>			
												к	о	л	о	к	о	л	_		о	к	о	1,<0,3>			
								к	о	л	о	к	о	л	_		о	к	о	л	о	_		0,'_'			
							к	о	л	о	к	о	л	_		о	к	о	л	о	_		к	1,<3,4>			
				к	о	л	о	к	о	л	_		о	к	о	л	о	_		к	о	л	о	к	1,<1,1>		
			к	о	л	о	к	о	л	_		о	к	о	л	о	_		к	о	л	о	к	о	1,<7,1>		
		к	о	л	о	к	о	л	_		о	к	о	л	о	_		к	о	л	о	к	о	л	1,<0,7>		
о	л	_		о	к	о	л	о	_		к	о	л	о	к	о	л	ь		н	и	:)			0,'ь'	
л	_		о	к	о	л	о	_		к	о	л	о	к	о	л	ь		н	и	:)			0,'н'		
_		о	к	о	л	о	_		к	о	л	о	к	о	л	ь		н	и	:)					0,'и'	

- Алгоритмы LZ77 и LZSS обладают следующими очевидными недостатками
 - невозможность кодирования подстрок, отстоящих друг от друга на расстоянии, большем длины словаря
 - длина подстроки, которую можно закодировать, ограничена размером буфера
 - увеличение размера словаря и буфера приводит к снижению эффективности кодирования и резко увеличится время работы кодера.

Алгоритм LZ78

- Алгоритмы этой группы не используют скользящего окна и в словарь помещают не все встречаемые при кодировании строки, а лишь "перспективные" с точки зрения вероятности последующего использования.

- На каждом шаге в словарь вставляется новая фраза, которая представляет собой сцепление (конкатенацию) одной из фраз S словаря, имеющей самое длинное совпадение со строкой буфера, и символа s .
- Символ s является символом, следующим за строкой буфера, для которой найдена совпадающая фраза S . В отличие от семейства LZ77 в словаре не может быть одинаковых фраз

- Алгоритм генерирует код, состоящий из индекса строки S в словаре и кода символа s .
- Затем строка Ss добавляется в словарь.
- Если словарь уже заполнен, то из него предварительно удаляют менее всех используемую в сравнениях строку.

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZ78
0	“ ”	--
1	К	<0,К>
2	О	<0,О>
3	Л	<0,Л>
4	ОК	<2,К>
5	ОЛ	<2,Л>
6	_	<0, _>
7	ОКО	<4,О>
8	ЛО	<3,О>
9	_К	<6,К>
10	ОЛО	<5,О>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZ78
0	“ ”	--
1	К	<0,К>
2	О	<0,О>
3	Л	<0,Л>
4	ОК	<2,К>
5	ОЛ	<2,Л>
6	_	<0,_>
7	ОКО	<4,О>
8	ЛО	<3,О>
9	_К	<6,К>
10	ОЛО	<5,О>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZ78
0	“ ”	--
1	К	<0,К>
2	О	<0,О>
3	Л	<0,Л>
4	ОК	<2,К>
5	ОЛ	<2,Л>
6	_	<0,_>
7	ОКО	<4,О>
8	ЛО	<3,О>
9	_К	<6,К>
10	ОЛО	<5,О>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZ78
0	“ ”	--
1	К	<0,К>
2	О	<0,О>
3	Л	<0,Л>
4	ОК	<2,К>
5	ОЛ	<2,Л>
6	_	<0, _>
7	ОКО	<4,О>
8	ЛО	<3,О>
9	_К	<6,К>
10	ОЛО	<5,О>

КОЛО**О****Л**_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZ78
0	“ ”	--
1	К	<0,К>
2	О	<0,О>
3	Л	<0,Л>
4	ОК	<2,К>
5	ОЛ	<2,Л>
6	_	<0, _>
7	ОКО	<4,О>
8	ЛО	<3,О>
9	_К	<6,К>
10	ОЛО	<5,О>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZ78
0	“ ”	--
1	К	<0,К>
2	О	<0,О>
3	Л	<0,Л>
4	ОК	<2,К>
5	ОЛ	<2,Л>
6	_	<0, _>
7	ОКО	<4,О>
8	ЛО	<3,О>
9	_К	<6,К>
10	ОЛО	<5,О>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZ78
0	“ ”	--
1	К	<0,К>
2	О	<0,О>
3	Л	<0,Л>
4	ОК	<2,К>
5	ОЛ	<2,Л>
6	_	<0, _>
7	ОКО	<4,О>
8	ЛО	<3,О>
9	_К	<6,К>
10	ОЛО	<5,О>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZ78
0	“ ”	--
1	К	<0,К>
2	О	<0,О>
3	Л	<0,Л>
4	ОК	<2,К>
5	ОЛ	<2,Л>
6	_	<0, _>
7	ОКО	<4,О>
8	ЛО	<3,О>
9	_К	<6,К>
10	ОЛО	<5,О>

КОЛОКОЛ_ОКОЛО_**К**ОЛОКОЛЬНИ:)

№	Словарь	Код LZ78
0	“ ”	--
1	К	<0,К>
2	О	<0,О>
3	Л	<0,Л>
4	ОК	<2,К>
5	ОЛ	<2,Л>
6	<u> </u>	<0,_ <u> </u> >
7	ОКО	<4,О>
8	ЛО	<3,О>
9	<u> </u> К	<6,К>
10	ОЛО	<5,О>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZ78
0	“ ”	--
1	К	<0,К>
2	О	<0,О>
3	Л	<0,Л>
4	ОК	<2,К>
5	ОЛ	<2,Л>
6	_	<0,_>
7	ОКО	<4,О>
8	ЛО	<3,О>
9	_К	<6,К>
10	ОЛО	<5,О>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZ78
11	КО	<1,О>
12	ЛЬ	<3,Ь>
13	Н	<0,Н>
14	И	<0,И>
15	:	<0,:>

КОЛОКОЛ_ОКОЛО_КОЛОКО**Л****Ь**НИ:)

№	Словарь	Код LZ78
11	КО	<1,О>
12	ЛЬ	<3,Ь>
13	Н	<0,Н>
14	И	<0,И>
15	:	<0,:>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬ**Н**И:)

№	Словарь	Код LZ78
11	КО	<1,О>
12	ЛЬ	<3,Ь>
13	Н	<0,Н>
14	И	<0,И>
15	:	<0,:>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:

№	Словарь	Код LZ78
11	КО	<1,О>
12	ЛЬ	<3,Ь>
13	Н	<0,Н>
14	И	<0,И>
15	:	<0,:>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZ78
11	КО	<1,О>
12	ЛЬ	<3,Ь>
13	Н	<0,Н>
14	И	<0,И>
15	:	<0,:>

Алгоритм LZW

- **Алгоритм LZW предложен Т. Уэлчем (Terry Welch) в 1984 г. как модификация алгоритма LZ78.**

- **Словарь инициализируется всеми возможными односимвольными фразами (обычно 256 символов)**
- **Алгоритм генерирует код, состоящий из номера строки, которая максимально совпадает с фрагментом сообщения.**
- **Затем новая строка добавляется в словарь. Если словарь уже заполнен, то из него предварительно удаляют менее всех используемую в сравнениях строку.**

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZW
0-255	ASCII	--
256	КО	<0,'K'>
257	ОЛ	<0,'Л'>
258	ЛО	<0,'О'>
259	ОК	<0,'K'>
260	КОЛ	<256>
261	Л_	<0,'Л'>
262	_О	<0,'_'>
263	ОКО	<259>
264	ОЛО	<257>
265	О_	<0,'О'>
266	_К	<0,'_'>
277	КОЛО	<260>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZW
0-255	ASCII	--
256	КО	<0,'K'>
257	ОЛ	<0,'O'>
258	ЛО	<0,'Л'>
259	ОК	<0,'O'>
260	КОЛ	<256>
261	Л_	<0,'Л'>
262	_О	<0,'_'>
263	ОКО	<259>
264	ОЛО	<257>
265	О_	<0,'O'>
266	_К	<0,'_'>
277	КОЛО	<260>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZW
0-255	ASCII	--
256	КО	<0,'K'>
257	ОЛ	<0,'O'>
258	ЛО	<0,'Л'>
259	ОК	<0,'O'>
260	КОЛ	<256>
261	Л_	<0,'Л'>
262	_О	<0,'_'>
263	ОКО	<259>
264	ОЛО	<257>
265	О_	<0,'O'>
266	_К	<0,'_'>
277	КОЛО	<260>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZW
0-255	ASCII	--
256	КО	<0,'K'>
257	ОЛ	<0,'O'>
258	ЛО	<0,'Л'>
259	ОК	<0,'O'>
260	КОЛ	<256>
261	Л_	<0,'Л'>
262	_О	<0,'_'>
263	ОКО	<259>
264	ОЛО	<257>
265	О_	<0,'O'>
266	_К	<0,'_'>
277	КОЛО	<260>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZW
0-255	ASCII	--
256	КО	<0,'K'>
257	ОЛ	<0,'O'>
258	ЛО	<0,'Л'>
259	ОК	<0,'O'>
260	КОЛ	<256>
261	Л_	<0,'Л'>
262	_О	<0,'_'>
263	ОКО	<259>
264	ОЛО	<257>
265	О_	<0,'O'>
266	_К	<0,'_'>
277	КОЛО	<260>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZW
0-255	ASCII	--
256	КО	<0,'K'>
257	ОЛ	<0,'Л'>
258	ЛО	<0,'O'>
259	ОК	<0,'K'>
260	КОЛ	<256>
261	Л_	<0,'Л'>
262	_О	<0,'_'>
263	ОКО	<259>
264	ОЛО	<257>
265	О_	<0,'O'>
266	_К	<0,'_'>
277	КОЛО	<260>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZW
0-255	ASCII	--
256	КО	<0,'K'>
257	ОЛ	<0,'Л'>
258	ЛО	<0,'О'>
259	ОК	<0,'K'>
260	КОЛ	<256>
261	Л_	<0,'Л'>
262	_О	<0,'_'>
263	ОКО	<259>
264	ОЛО	<257>
265	О_	<0,'О'>
266	_К	<0,'_'>
277	КОЛО	<260>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZW
0-255	ASCII	--
256	КО	<0,'K'>
257	ОЛ	<0,'Л'>
258	ЛО	<0,'О'>
259	ОК	<0,'K'>
260	КОЛ	<256>
261	Л_	<0,'Л'>
262	_О	<0,'_'>
263	ОКО	<259>
264	ОЛО	<257>
265	О_	<0,'О'>
266	_К	<0,'_'>
277	КОЛО	<260>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZW
0-255	ASCII	--
256	КО	<0,'K'>
257	ОЛ	<0,'Л'>
258	ЛО	<0,'O'>
259	ОК	<0,'K'>
260	КОЛ	<256>
261	Л_	<0,'Л'>
262	_О	<0,'_'>
263	ОКО	<259>
264	ОЛО	<257>
265	О_	<0,'O'>
266	_К	<0,'_'>
277	КОЛО	<260>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZW
0-255	ASCII	--
256	КО	<0,'K'>
257	ОЛ	<0,'Л'>
258	ЛО	<0,'О'>
259	ОК	<0,'K'>
260	КОЛ	<256>
261	Л_	<0,'Л'>
262	_О	<0,'_'>
263	ОКО	<259>
264	ОЛО	<257>
265	О_	<0,'О'>
266	_К	<0,'_'>
277	КОЛО	<260>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZW
0-255	ASCII	--
256	КО	<0,'K'>
257	ОЛ	<0,'Л'>
258	ЛО	<0,'О'>
259	ОК	<0,'K'>
260	КОЛ	<256>
261	Л_	<0,'Л'>
262	_О	<0,'_'>
263	ОКО	<259>
264	ОЛО	<257>
265	О_	<0,'О'>
266	_К	<0,'_'>
277	КОЛО	<260>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZW
0-255	ASCII	--
256	КО	<0,'K'>
257	ОЛ	<0,'Л'>
258	ЛО	<0,'О'>
259	ОК	<0,'K'>
260	КОЛ	<256>
261	Л_	<0,'Л'>
262	_О	<0,'_'>
263	ОКО	<259>
264	ОЛО	<257>
265	О_	<0,'О'>
266	_К	<0,'_'>
277	КОЛО	<260>

КОЛОКОЛ_ОКОЛО_КОЛОКОЛЬНИ:)

№	Словарь	Код LZW
277	ОКОЛ	<263>
278	ЛЬ	<0,'Л'>
279	ЬН	<0,'Ь'>
280	НИ	<0,'Н'>
281	И:	<0,'И'>
282	:)	<0,': '>
283)	<0,') '>
284		
285		
286		
287		
288		
289		

- Как и в алгоритме LZ78, длина получаемых кодов LZW зависит от размера словаря.
- LZW-коды имеют постоянную длину (округленный двоичный логарифм размера словаря)

- При переполнении словаря, удаляют либо наиболее редко используемую фразу, либо все фразы отличающиеся от одиночного символа