LZ78

- In LZ78, no search buffer or sliding window are used
- Instead, a dictionary of previously encountered strings is used
- This dictionary starts empty, except from a null string at position zero
- The dictionary size is limited
- At each step during compression, the dictionary is searched for the longest sub-string which is a *suffix* of the yet to be encoded sequence
- The LZ78 encoder produces two-part token
  - □ *a pointer to a dictionary entry*
  - □ *the code of the next symbol*
- The *length* part has been eliminated, since it is implied in the dictionary
- Each token corresponds to a sub-string of encoded input symbols is added to the dictionary ***after*** the token is written on the compressed stream

Encoding of "*sir sid eastman easily teases sea sick seal*" using LZ8

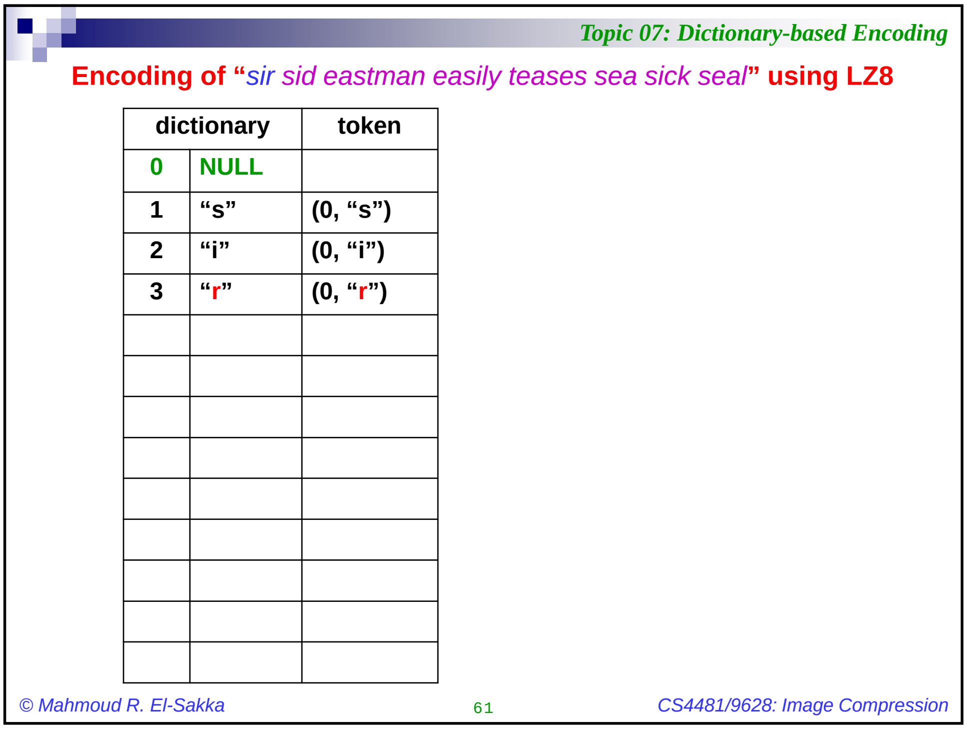
dictionary		token
0	NULL	

Encoding of “*sir sid eastman easily teases sea sick seal*” using LZ8

dictionary		token
0	NULL	
1	“s”	(0, “s”)

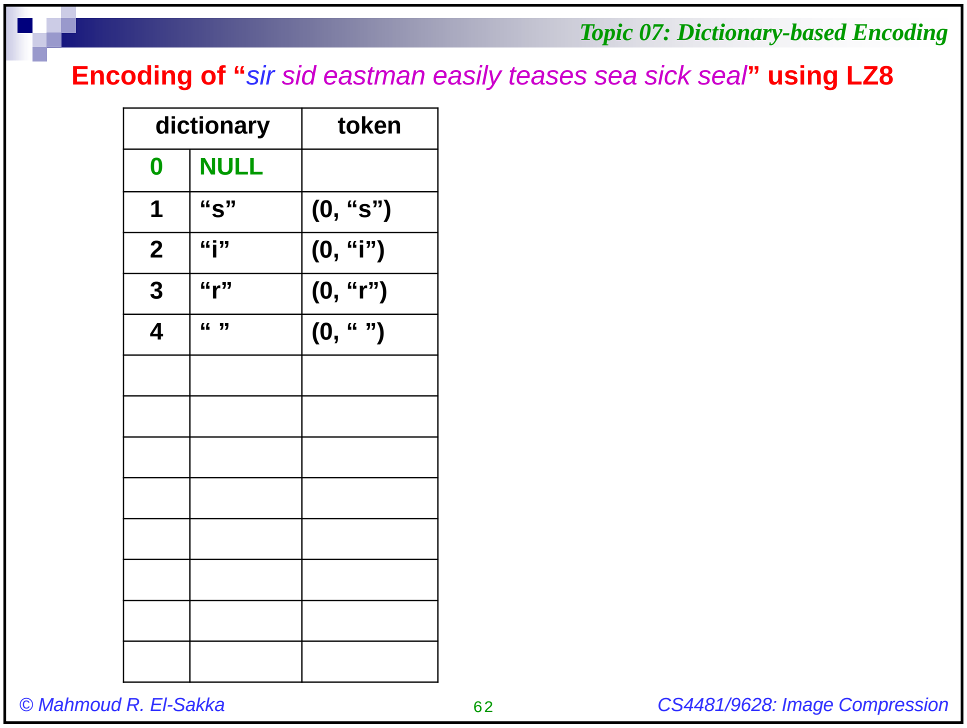
Encoding of “*sir sid eastman easily teases sea sick seal*” using LZ8

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)



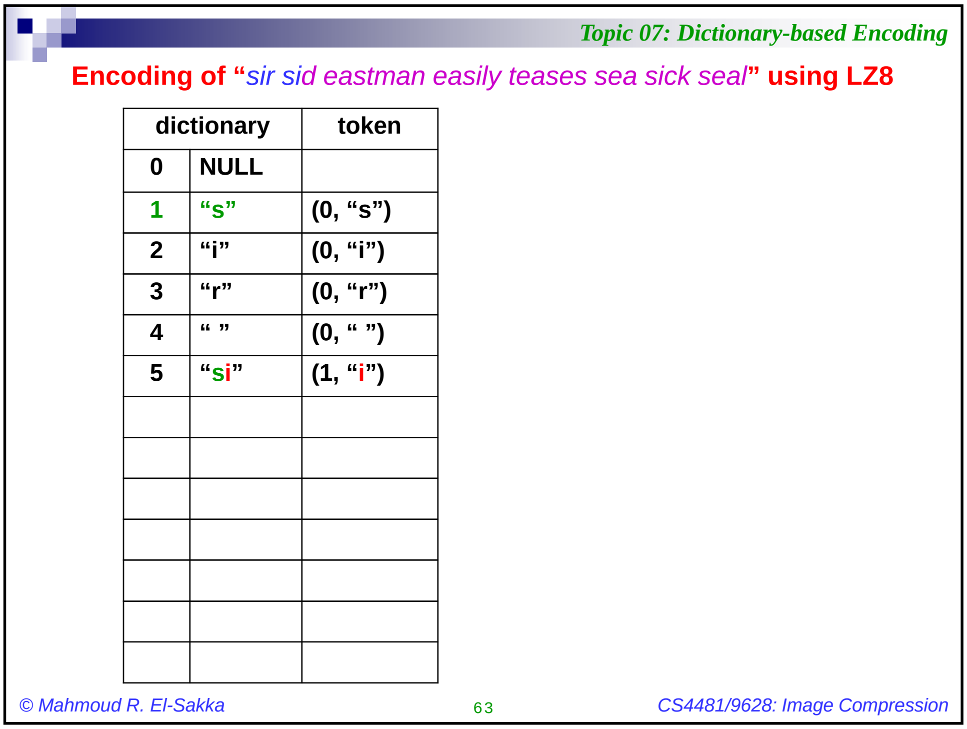
Encoding of "*sir sid eastman easily teases sea sick seal*" using LZ8

dictionary		token
0	NULL	
1	"s"	(0, "s")
2	"i"	(0, "i")
3	"r"	(0, "r")



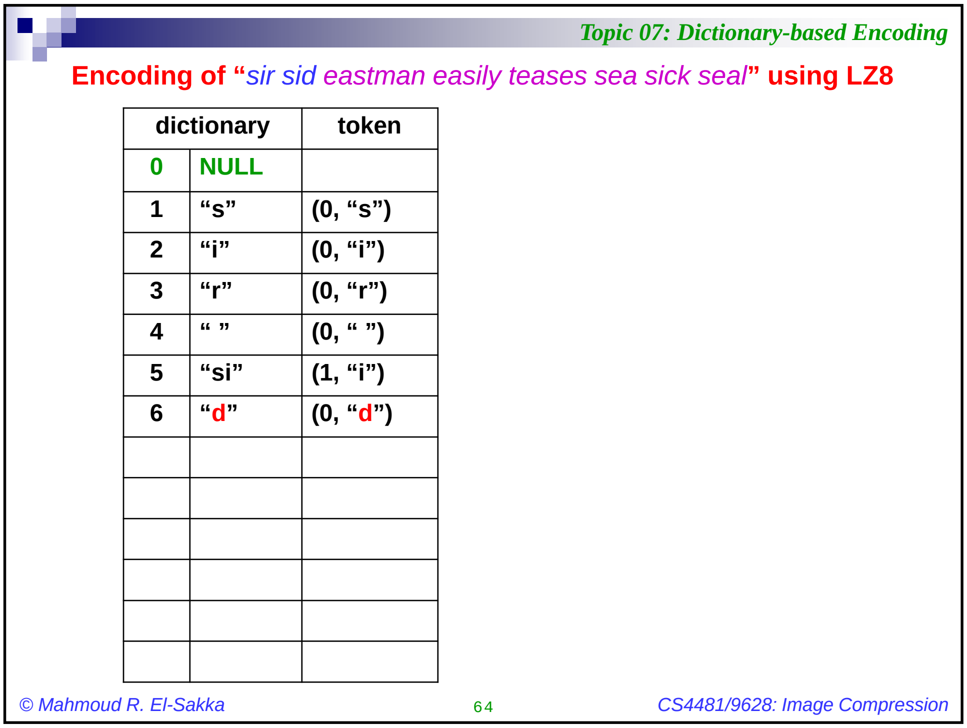
Encoding of "*sir sid eastman easily teases sea sick seal*" using LZ8

dictionary		token
0	NULL	
1	"s"	(0, "s")
2	"i"	(0, "i")
3	"r"	(0, "r")
4	" "	(0, " ")



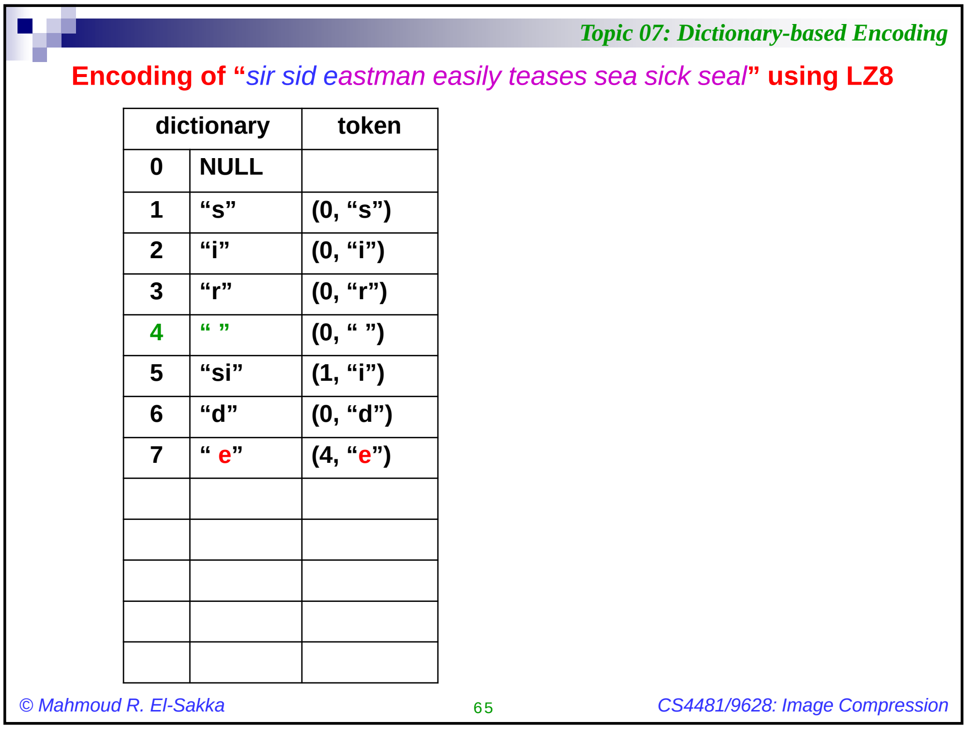
Encoding of "*sir sid eastman easily teases sea sick seal*" using LZ8

dictionary		token
0	NULL	
1	"s"	(0, "s")
2	"i"	(0, "i")
3	"r"	(0, "r")
4	" "	(0, " ")
5	"si"	(1, "i")



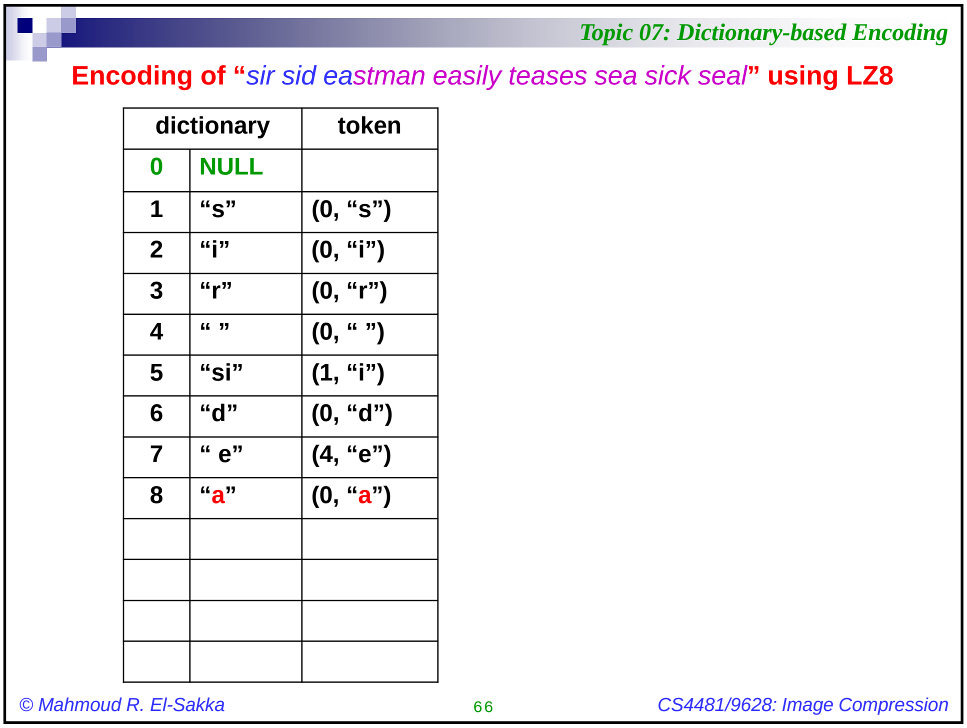
Encoding of "*sir sid eastman easily teases sea sick seal*" using LZ8

dictionary		token
0	NULL	
1	"s"	(0, "s")
2	"i"	(0, "i")
3	"r"	(0, "r")
4	" "	(0, " ")
5	"si"	(1, "i")
6	"d"	(0, "d")



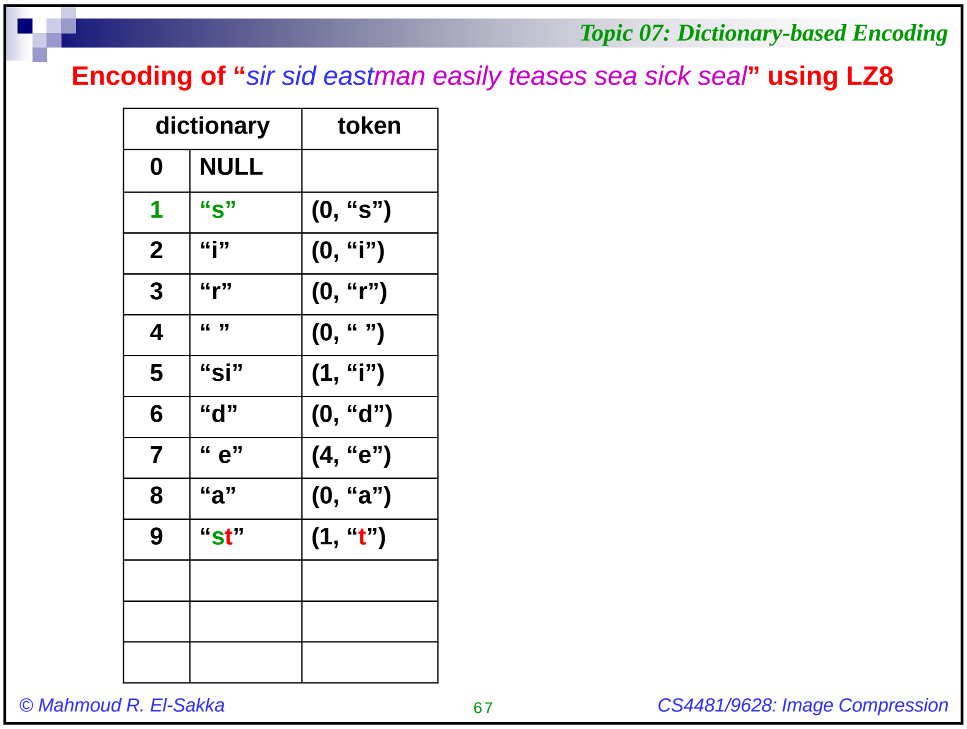
Encoding of "*sir sid eastman easily teases sea sick seal*" using LZ8

dictionary		token
0	NULL	
1	"s"	(0, "s")
2	"i"	(0, "i")
3	"r"	(0, "r")
4	" "	(0, " ")
5	"si"	(1, "i")
6	"d"	(0, "d")
7	" e"	(4, "e")



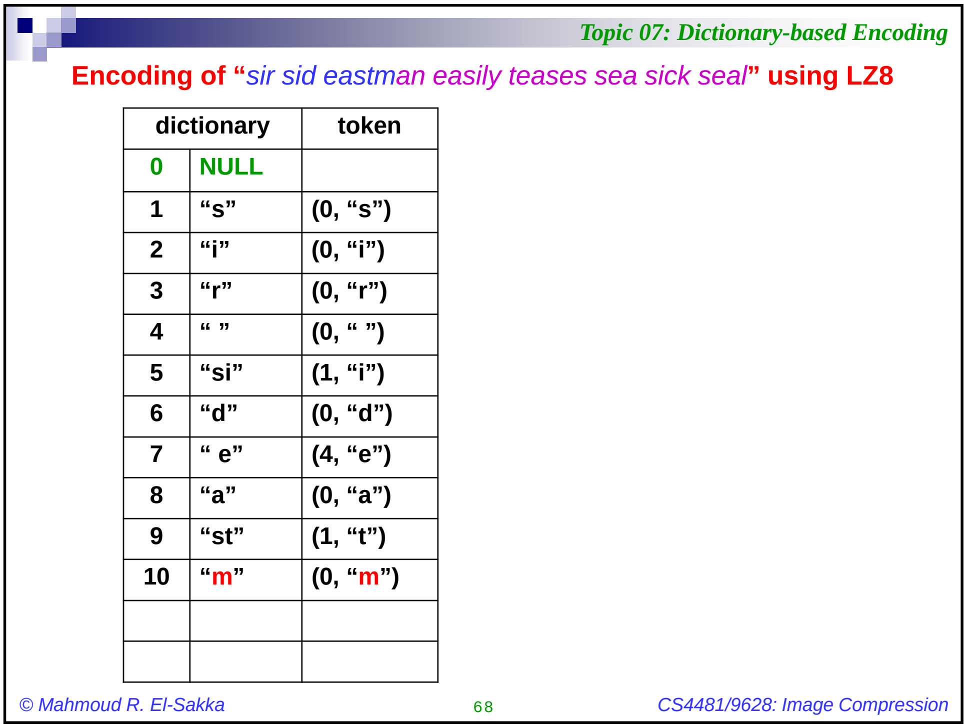
Encoding of "*sir sid eastman easily teases sea sick seal*" using LZ8

dictionary		token
0	NULL	
1	"s"	(0, "s")
2	"i"	(0, "i")
3	"r"	(0, "r")
4	" "	(0, " ")
5	"si"	(1, "i")
6	"d"	(0, "d")
7	" e"	(4, "e")
8	"a"	(0, "a")



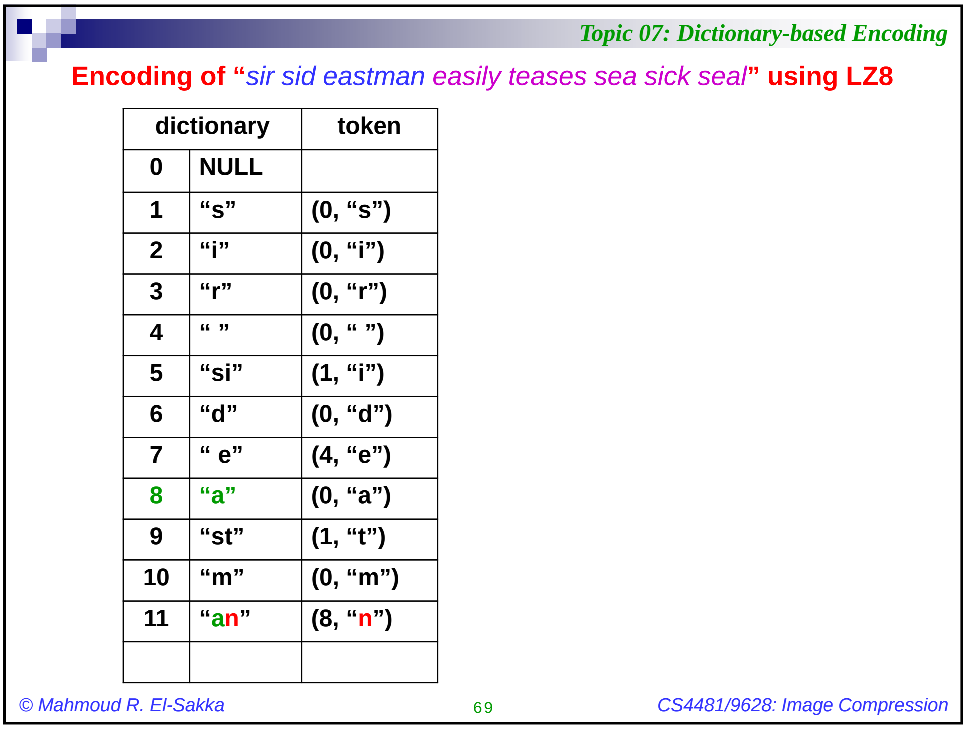
Encoding of "*sir sid eastman easily teases sea sick seal*" using LZ8

dictionary		token
0	NULL	
1	"s"	(0, "s")
2	"i"	(0, "i")
3	"r"	(0, "r")
4	" "	(0, " ")
5	"si"	(1, "i")
6	"d"	(0, "d")
7	" e"	(4, "e")
8	"a"	(0, "a")
9	"st"	(1, "t")



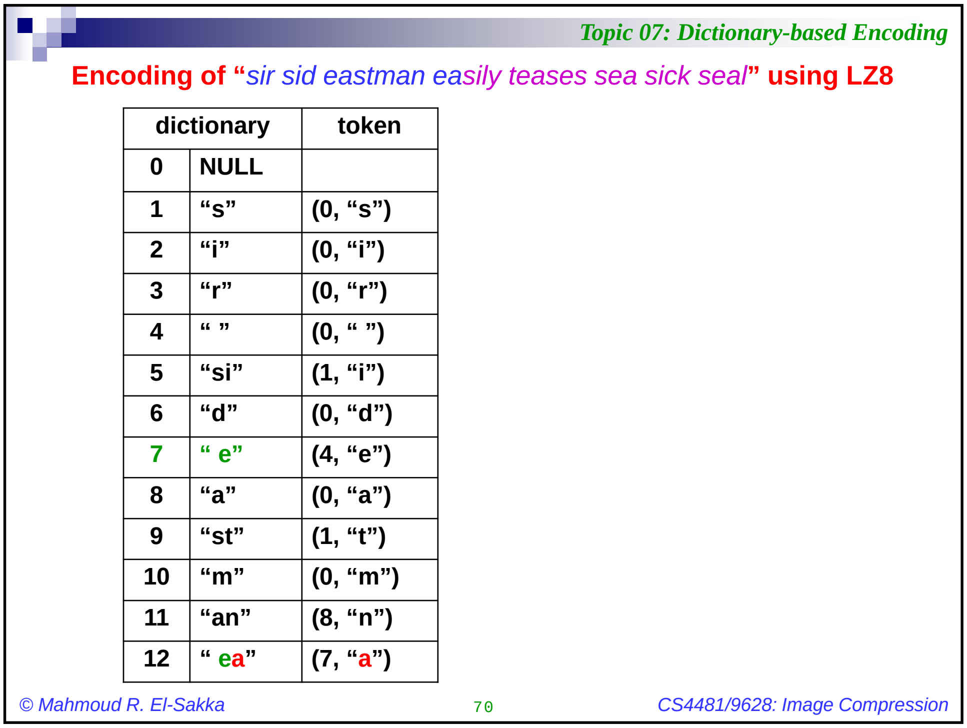
Encoding of "*sir sid eastman easily teases sea sick seal*" using LZ8

dictionary		token
0	NULL	
1	"s"	(0, "s")
2	"i"	(0, "i")
3	"r"	(0, "r")
4	" "	(0, " ")
5	"si"	(1, "i")
6	"d"	(0, "d")
7	" e"	(4, "e")
8	"a"	(0, "a")
9	"st"	(1, "t")
10	"m"	(0, "m")



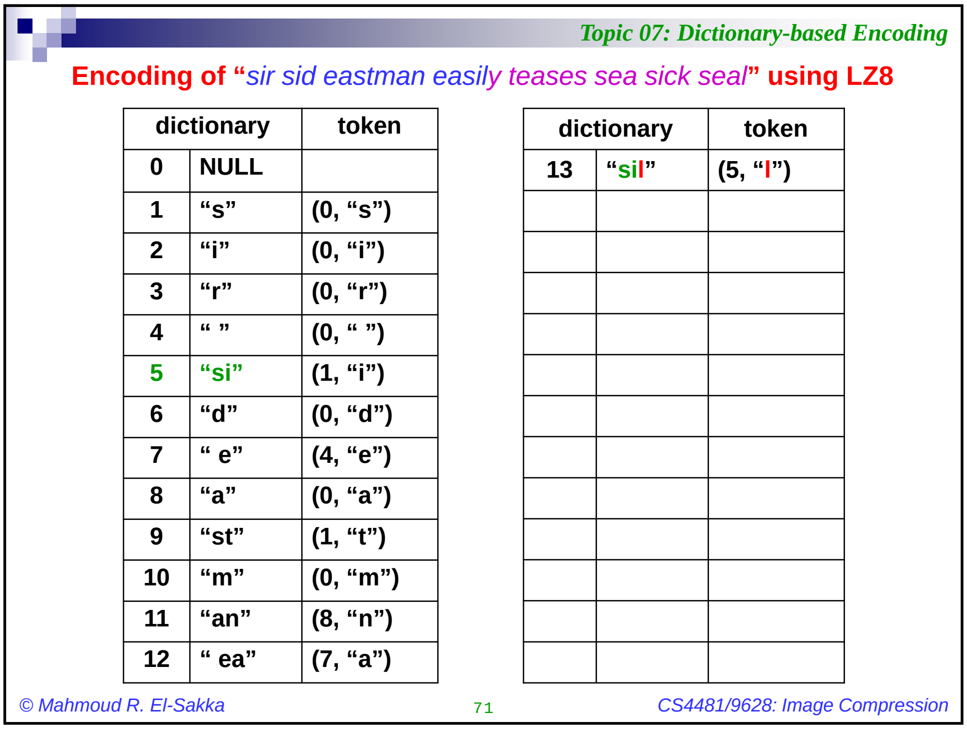
Encoding of "*sir sid eastman easily teases sea sick seal*" using LZ8

dictionary		token
0	NULL	
1	"s"	(0, "s")
2	"i"	(0, "i")
3	"r"	(0, "r")
4	" "	(0, " ")
5	"si"	(1, "i")
6	"d"	(0, "d")
7	" e"	(4, "e")
8	"a"	(0, "a")
9	"st"	(1, "t")
10	"m"	(0, "m")
11	"an"	(8, "n")



Encoding of "*sir sid eastman easily teases sea sick seal*" using LZ8

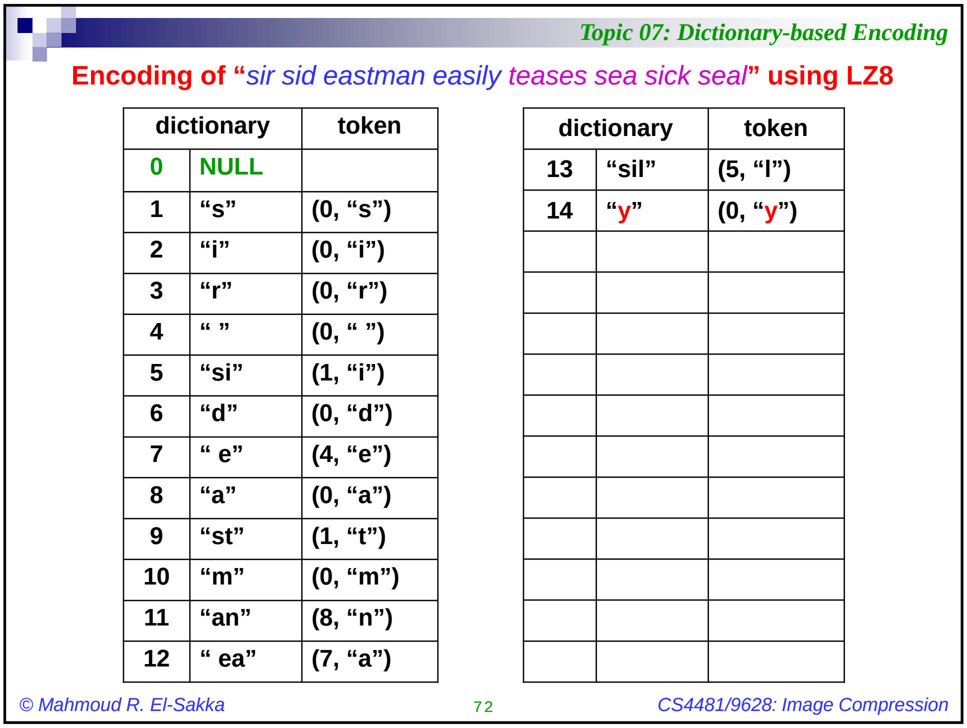
dictionary		token
0	NULL	
1	"s"	(0, "s")
2	"i"	(0, "i")
3	"r"	(0, "r")
4	" "	(0, " ")
5	"si"	(1, "i")
6	"d"	(0, "d")
7	" e"	(4, "e")
8	"a"	(0, "a")
9	"st"	(1, "t")
10	"m"	(0, "m")
11	"an"	(8, "n")
12	" ea"	(7, "a")



Encoding of “*sir sid eastman easily teases sea sick seal*” using LZ8

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

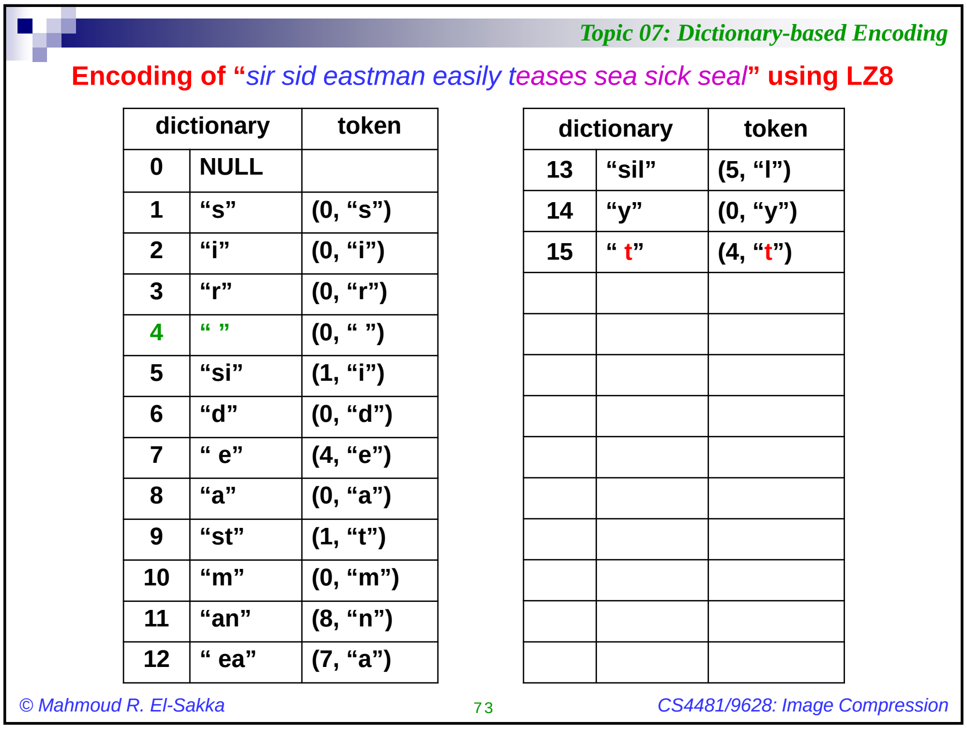
dictionary		token
13	“sil”	(5, “l”)



Encoding of “*sir sid eastman easily teases sea sick seal*” using LZ8

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

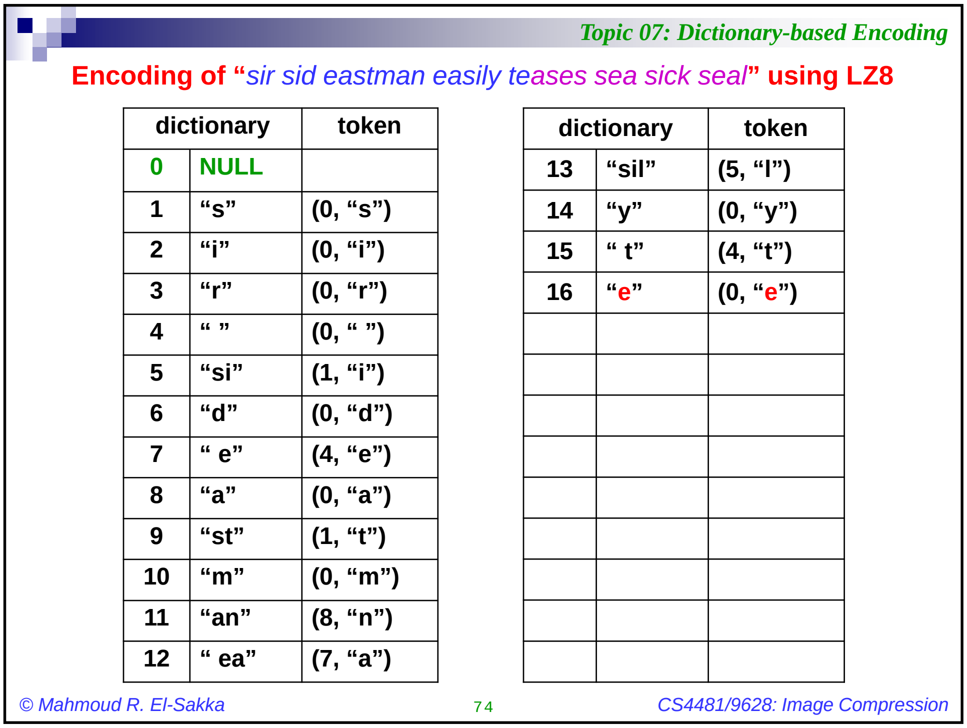
dictionary		token
13	“sil”	(5, “l”)
14	“y”	(0, “y”)



Encoding of “*sir sid eastman easily teases sea sick seal*” using LZ8

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

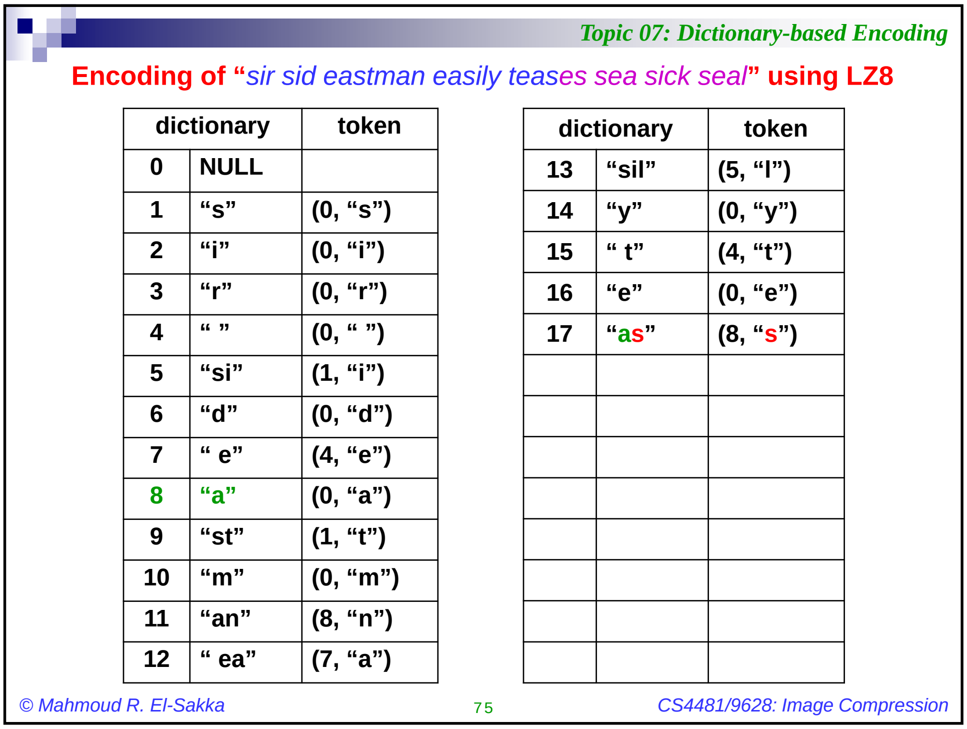
dictionary		token
13	“sil”	(5, “l”)
14	“y”	(0, “y”)
15	“ t”	(4, “t”)



Encoding of “*sir sid eastman easily teases sea sick seal*” using LZ8

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

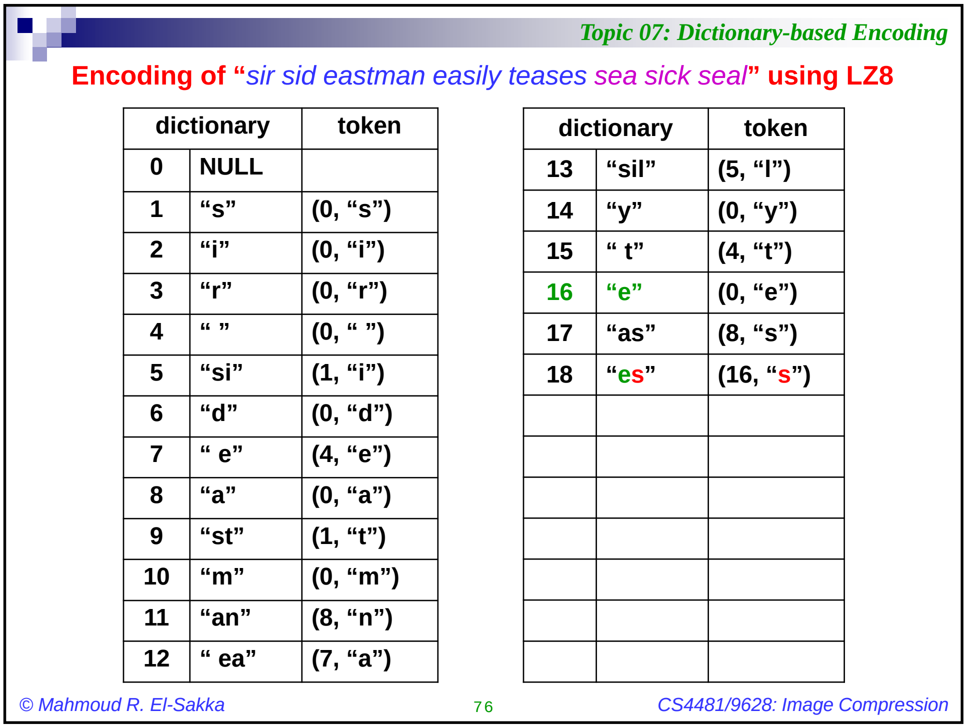
dictionary		token
13	“sil”	(5, “l”)
14	“y”	(0, “y”)
15	“ t”	(4, “t”)
16	“e”	(0, “e”)



Encoding of "*sir sid eastman easily teases sea sick seal*" using LZ8

dictionary		token
0	NULL	
1	"s"	(0, "s")
2	"i"	(0, "i")
3	"r"	(0, "r")
4	" "	(0, " ")
5	"si"	(1, "i")
6	"d"	(0, "d")
7	" e"	(4, "e")
8	"a"	(0, "a")
9	"st"	(1, "t")
10	"m"	(0, "m")
11	"an"	(8, "n")
12	" ea"	(7, "a")

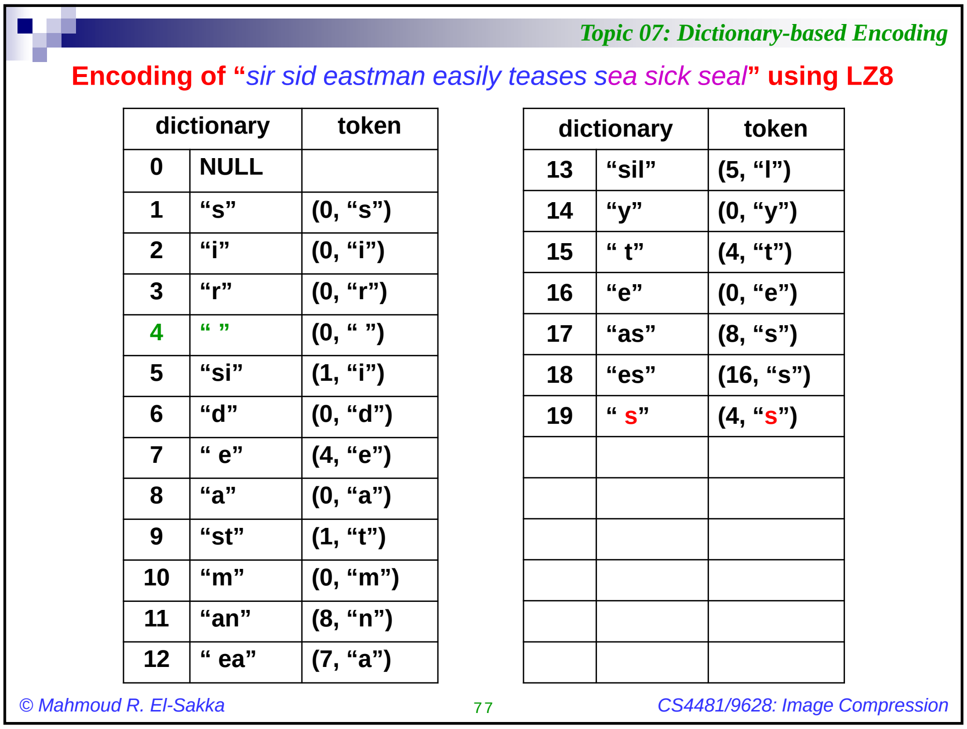
dictionary		token
13	"sil"	(5, "l")
14	"y"	(0, "y")
15	" t"	(4, "t")
16	"e"	(0, "e")
17	"as"	(8, "s")



Encoding of "*sir sid eastman easily teases sea sick seal*" using LZ8

dictionary		token
0	NULL	
1	"s"	(0, "s")
2	"i"	(0, "i")
3	"r"	(0, "r")
4	" "	(0, " ")
5	"si"	(1, "i")
6	"d"	(0, "d")
7	" e"	(4, "e")
8	"a"	(0, "a")
9	"st"	(1, "t")
10	"m"	(0, "m")
11	"an"	(8, "n")
12	" ea"	(7, "a")

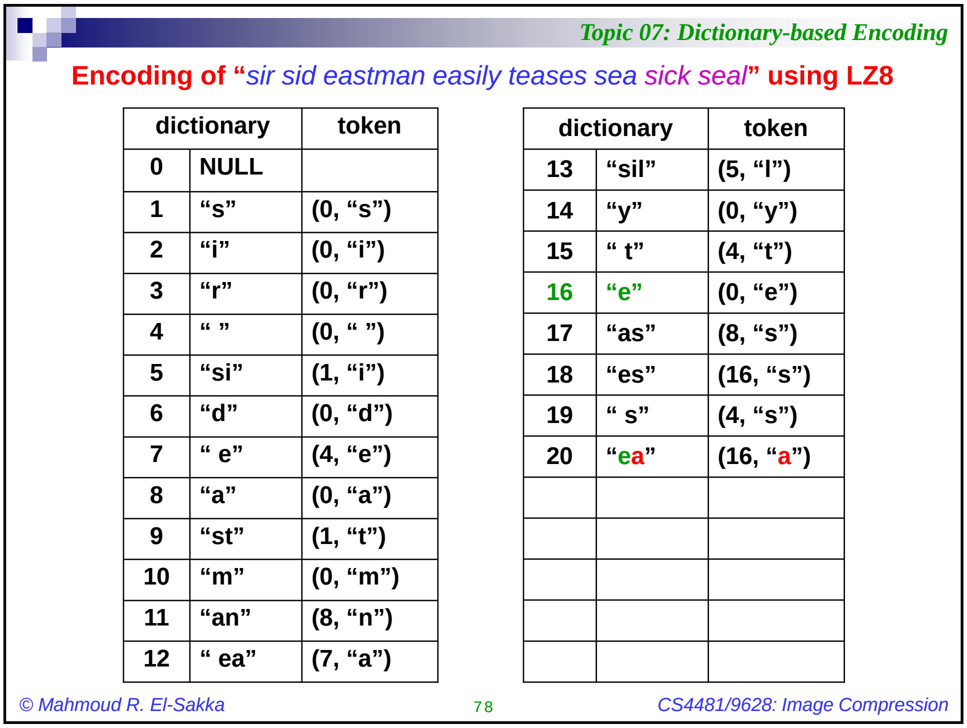
dictionary		token
13	"sil"	(5, "l")
14	"y"	(0, "y")
15	" t"	(4, "t")
16	"e"	(0, "e")
17	"as"	(8, "s")
18	"es"	(16, "s")



Encoding of "*sir sid eastman easily teases sea sick seal*" using LZ8

dictionary		token
0	NULL	
1	"s"	(0, "s")
2	"i"	(0, "i")
3	"r"	(0, "r")
4	" "	(0, " ")
5	"si"	(1, "i")
6	"d"	(0, "d")
7	" e"	(4, "e")
8	"a"	(0, "a")
9	"st"	(1, "t")
10	"m"	(0, "m")
11	"an"	(8, "n")
12	" ea"	(7, "a")

dictionary		token
13	"sil"	(5, "l")
14	"y"	(0, "y")
15	" t"	(4, "t")
16	"e"	(0, "e")
17	"as"	(8, "s")
18	"es"	(16, "s")
19	" s"	(4, "s")



Encoding of "*sir sid eastman easily teases sea sick seal*" using LZ8

dictionary		token
0	NULL	
1	"s"	(0, "s")
2	"i"	(0, "i")
3	"r"	(0, "r")
4	" "	(0, " ")
5	"si"	(1, "i")
6	"d"	(0, "d")
7	" e"	(4, "e")
8	"a"	(0, "a")
9	"st"	(1, "t")
10	"m"	(0, "m")
11	"an"	(8, "n")
12	" ea"	(7, "a")

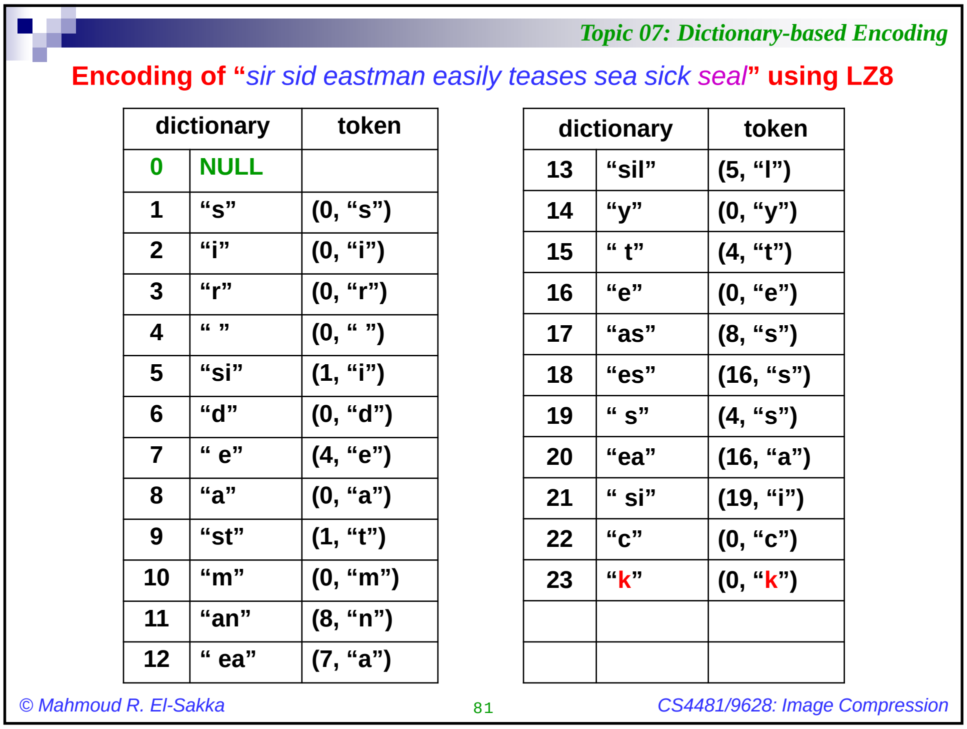
dictionary		token
13	"sil"	(5, "l")
14	"y"	(0, "y")
15	" t"	(4, "t")
16	"e"	(0, "e")
17	"as"	(8, "s")
18	"es"	(16, "s")
19	" s"	(4, "s")
20	"ea"	(16, "a")

Encoding of “*sir sid eastman easily teases sea sick seal*” using LZ8

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)
13	“sil”	(5, “l”)
14	“y”	(0, “y”)
15	“ t”	(4, “t”)
16	“e”	(0, “e”)
17	“as”	(8, “s”)
18	“es”	(16, “s”)
19	“ s”	(4, “s”)
20	“ea”	(16, “a”)
21	“ si”	(19, “i”)

Encoding of “*sir sid eastman easily teases sea sick seal*” using LZ8

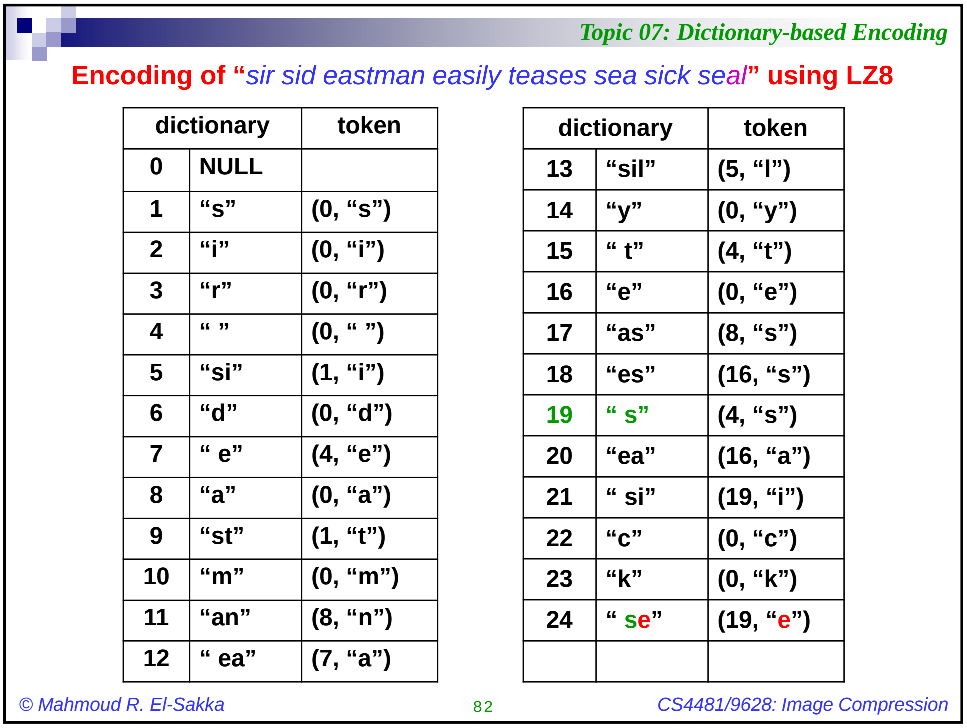
dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)
13	“sil”	(5, “l”)
14	“y”	(0, “y”)
15	“ t”	(4, “t”)
16	“e”	(0, “e”)
17	“as”	(8, “s”)
18	“es”	(16, “s”)
19	“ s”	(4, “s”)
20	“ea”	(16, “a”)
21	“ si”	(19, “i”)
22	“c”	(0, “c”)



Encoding of “*sir sid eastman easily teases sea sick seal*” using LZ8

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

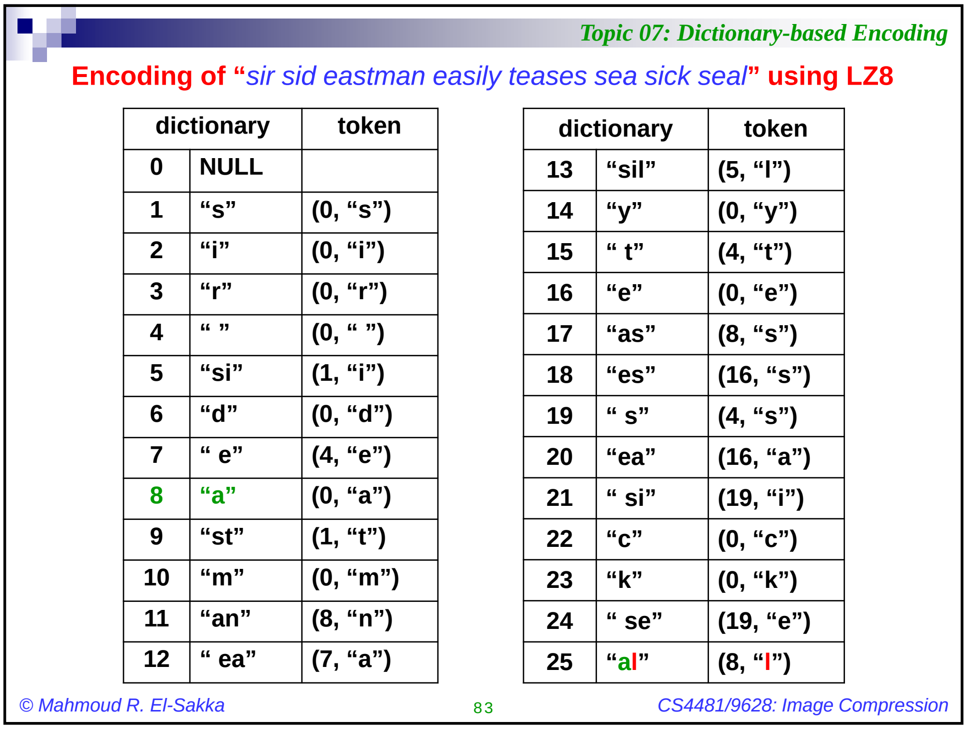
dictionary		token
13	“sil”	(5, “l”)
14	“y”	(0, “y”)
15	“ t”	(4, “t”)
16	“e”	(0, “e”)
17	“as”	(8, “s”)
18	“es”	(16, “s”)
19	“ s”	(4, “s”)
20	“ea”	(16, “a”)
21	“ si”	(19, “i”)
22	“c”	(0, “c”)
23	“k”	(0, “k”)



Encoding of “*sir sid eastman easily teases sea sick seal*” using LZ8

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

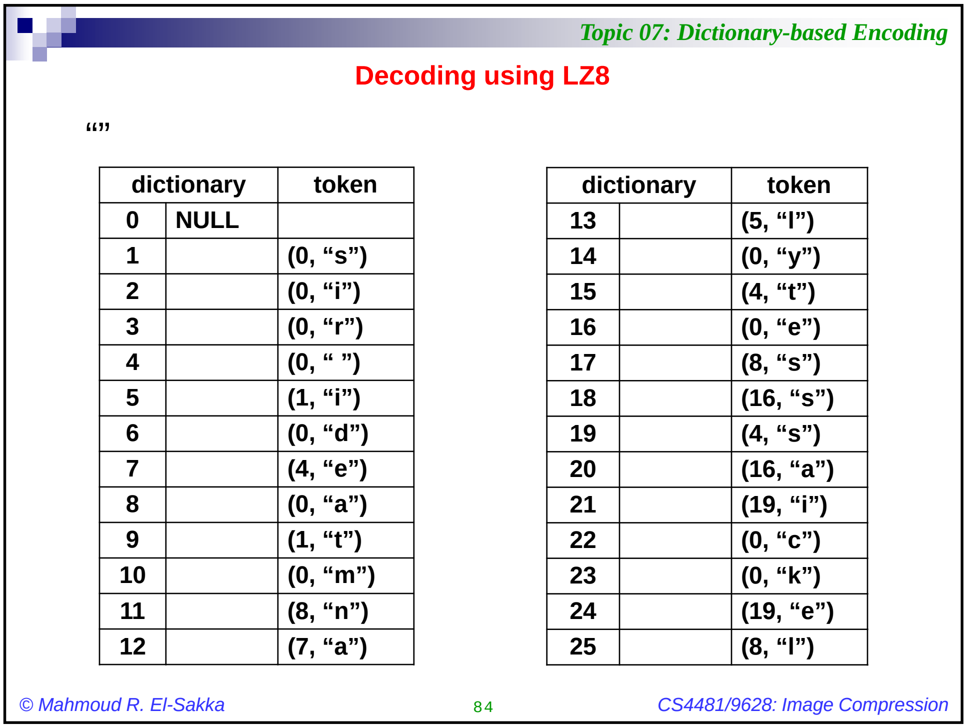
dictionary		token
13	“sil”	(5, “l”)
14	“y”	(0, “y”)
15	“ t”	(4, “t”)
16	“e”	(0, “e”)
17	“as”	(8, “s”)
18	“es”	(16, “s”)
19	“ s”	(4, “s”)
20	“ea”	(16, “a”)
21	“ si”	(19, “i”)
22	“c”	(0, “c”)
23	“k”	(0, “k”)
24	“ se”	(19, “e”)



Encoding of “*sir sid eastman easily teases sea sick seal*” using LZ8

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

dictionary		token
13	“sil”	(5, “l”)
14	“y”	(0, “y”)
15	“ t”	(4, “t”)
16	“e”	(0, “e”)
17	“as”	(8, “s”)
18	“es”	(16, “s”)
19	“ s”	(4, “s”)
20	“ea”	(16, “a”)
21	“ si”	(19, “i”)
22	“c”	(0, “c”)
23	“k”	(0, “k”)
24	“ se”	(19, “e”)
25	“al”	(8, “l”)

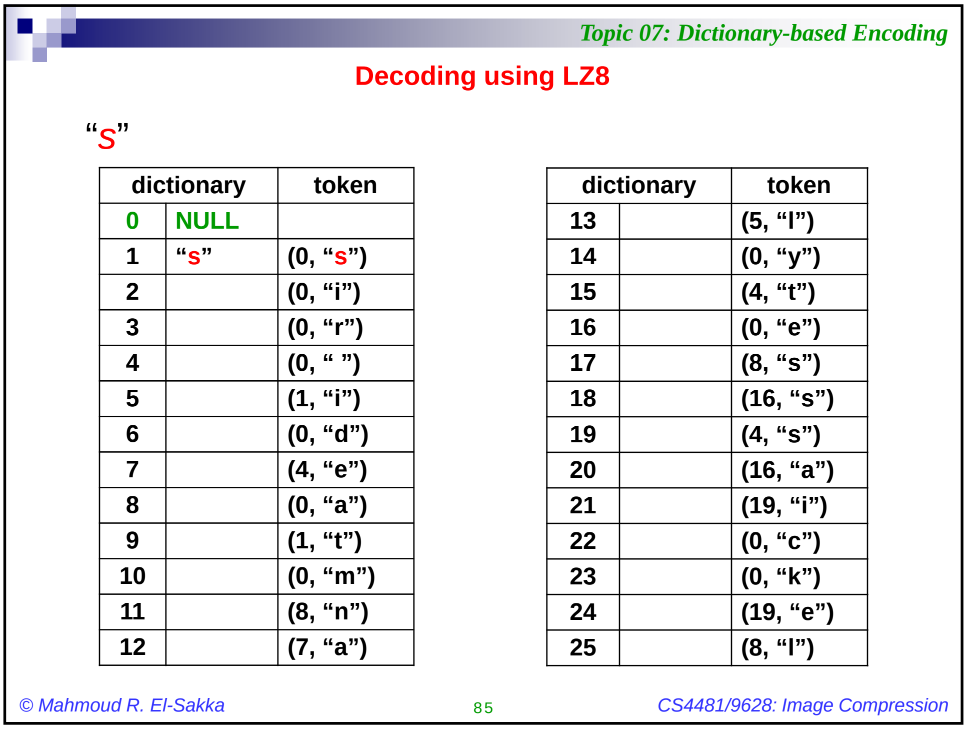


Decoding using LZ8

“”

dictionary		token
0	NULL	
1		(0, “s”)
2		(0, “i”)
3		(0, “r”)
4		(0, “ ”)
5		(1, “i”)
6		(0, “d”)
7		(4, “e”)
8		(0, “a”)
9		(1, “t”)
10		(0, “m”)
11		(8, “n”)
12		(7, “a”)

dictionary		token
13		(5, “l”)
14		(0, “y”)
15		(4, “t”)
16		(0, “e”)
17		(8, “s”)
18		(16, “s”)
19		(4, “s”)
20		(16, “a”)
21		(19, “i”)
22		(0, “c”)
23		(0, “k”)
24		(19, “e”)
25		(8, “l”)

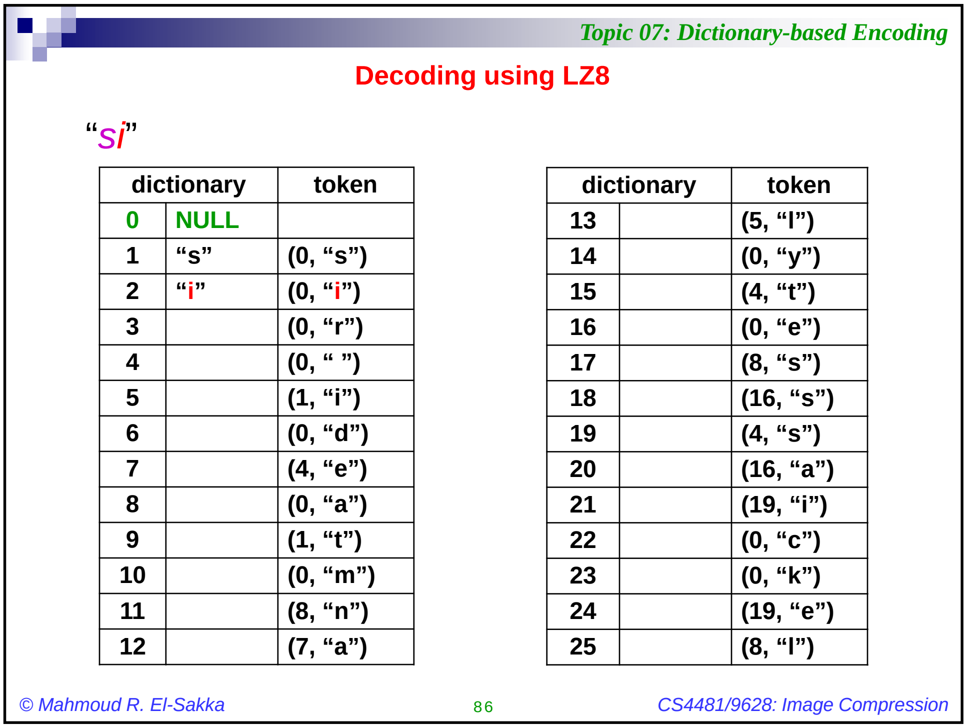


Decoding using LZ8

“*S*”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2		(0, “i”)
3		(0, “r”)
4		(0, “ ”)
5		(1, “i”)
6		(0, “d”)
7		(4, “e”)
8		(0, “a”)
9		(1, “t”)
10		(0, “m”)
11		(8, “n”)
12		(7, “a”)

dictionary		token
13		(5, “l”)
14		(0, “y”)
15		(4, “t”)
16		(0, “e”)
17		(8, “s”)
18		(16, “s”)
19		(4, “s”)
20		(16, “a”)
21		(19, “i”)
22		(0, “c”)
23		(0, “k”)
24		(19, “e”)
25		(8, “l”)

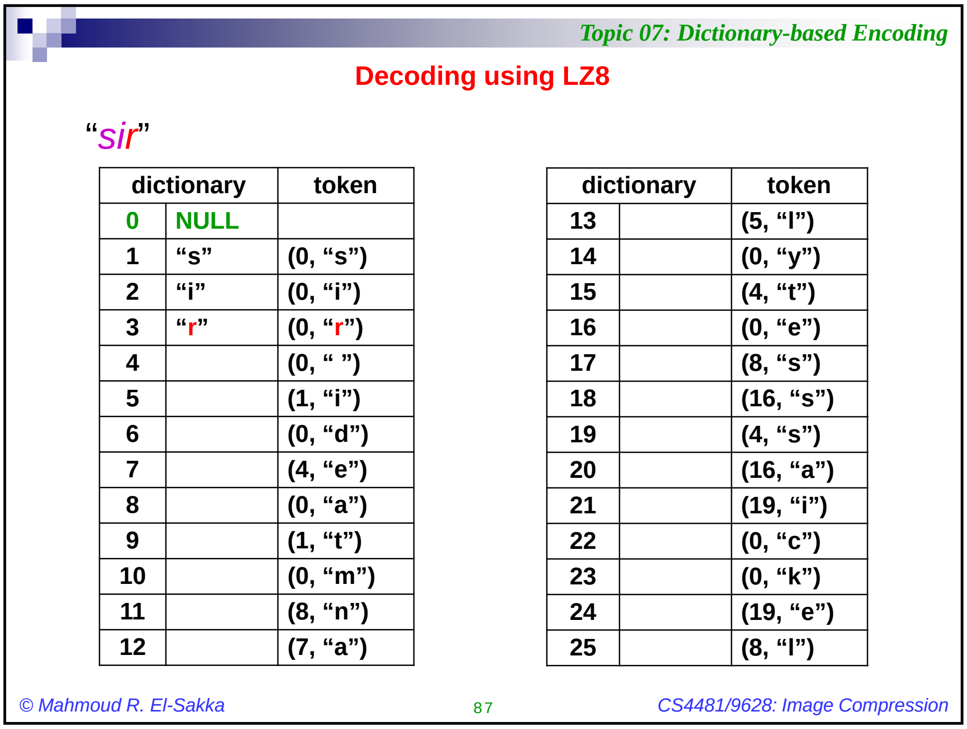


Decoding using LZ8

“*Si*”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3		(0, “r”)
4		(0, “ ”)
5		(1, “i”)
6		(0, “d”)
7		(4, “e”)
8		(0, “a”)
9		(1, “t”)
10		(0, “m”)
11		(8, “n”)
12		(7, “a”)

dictionary		token
13		(5, “l”)
14		(0, “y”)
15		(4, “t”)
16		(0, “e”)
17		(8, “s”)
18		(16, “s”)
19		(4, “s”)
20		(16, “a”)
21		(19, “i”)
22		(0, “c”)
23		(0, “k”)
24		(19, “e”)
25		(8, “l”)

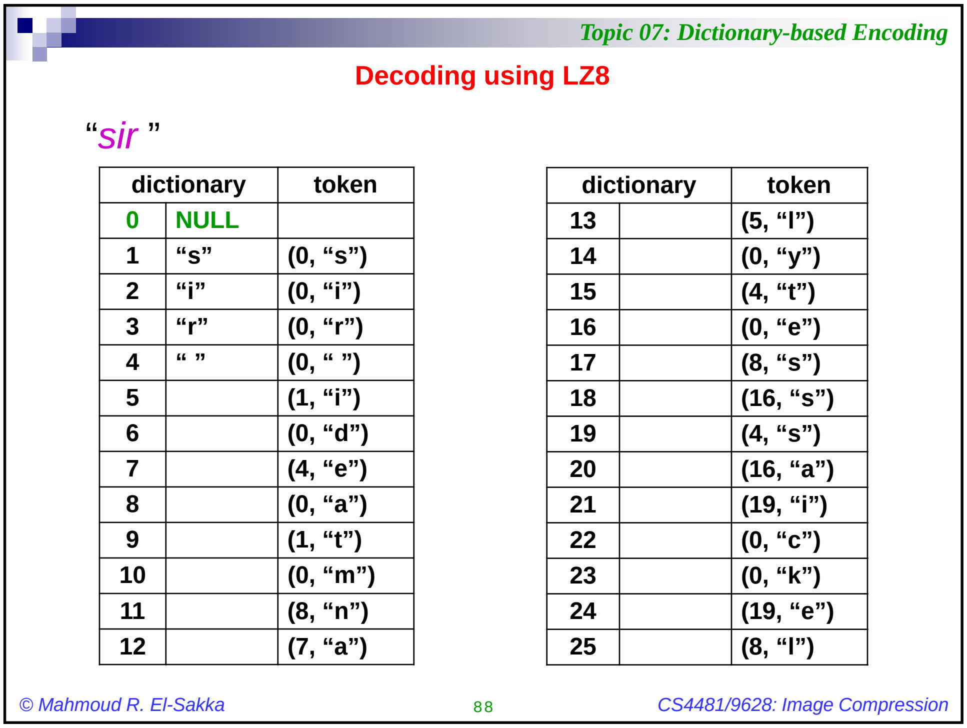


Decoding using LZ8

“*sir*”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4		(0, “ ”)
5		(1, “i”)
6		(0, “d”)
7		(4, “e”)
8		(0, “a”)
9		(1, “t”)
10		(0, “m”)
11		(8, “n”)
12		(7, “a”)

dictionary		token
13		(5, “l”)
14		(0, “y”)
15		(4, “t”)
16		(0, “e”)
17		(8, “s”)
18		(16, “s”)
19		(4, “s”)
20		(16, “a”)
21		(19, “i”)
22		(0, “c”)
23		(0, “k”)
24		(19, “e”)
25		(8, “l”)



Decoding using LZ8

“*sir* ”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5		(1, “i”)
6		(0, “d”)
7		(4, “e”)
8		(0, “a”)
9		(1, “t”)
10		(0, “m”)
11		(8, “n”)
12		(7, “a”)

dictionary		token
13		(5, “l”)
14		(0, “y”)
15		(4, “t”)
16		(0, “e”)
17		(8, “s”)
18		(16, “s”)
19		(4, “s”)
20		(16, “a”)
21		(19, “i”)
22		(0, “c”)
23		(0, “k”)
24		(19, “e”)
25		(8, “l”)

Decoding using LZ8

"sir si"

dictionary		token
0	NULL	
1	"s"	(0, "s")
2	"i"	(0, "i")
3	"r"	(0, "r")
4	" "	(0, " ")
5	"si"	(1, "i")
6		(0, "d")
7		(4, "e")
8		(0, "a")
9		(1, "t")
10		(0, "m")
11		(8, "n")
12		(7, "a")

dictionary		token
13		(5, "l")
14		(0, "y")
15		(4, "t")
16		(0, "e")
17		(8, "s")
18		(16, "s")
19		(4, "s")
20		(16, "a")
21		(19, "i")
22		(0, "c")
23		(0, "k")
24		(19, "e")
25		(8, "l")

Decoding using LZ8

"sir sid"

dictionary		token
0	NULL	
1	"s"	(0, "s")
2	"i"	(0, "i")
3	"r"	(0, "r")
4	" "	(0, " ")
5	"si"	(1, "i")
6	"d"	(0, "d")
7		(4, "e")
8		(0, "a")
9		(1, "t")
10		(0, "m")
11		(8, "n")
12		(7, "a")

dictionary		token
13		(5, "l")
14		(0, "y")
15		(4, "t")
16		(0, "e")
17		(8, "s")
18		(16, "s")
19		(4, "s")
20		(16, "a")
21		(19, "i")
22		(0, "c")
23		(0, "k")
24		(19, "e")
25		(8, "l")

Decoding using LZ8

“*sir sid e*”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8		(0, “a”)
9		(1, “t”)
10		(0, “m”)
11		(8, “n”)
12		(7, “a”)

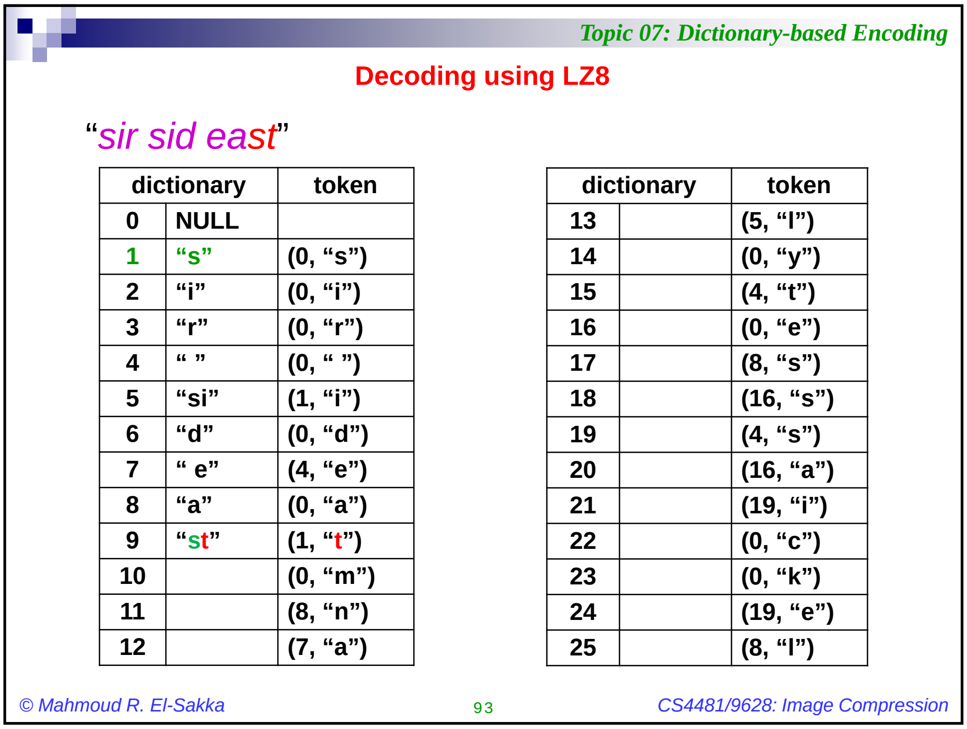
dictionary		token
13		(5, “l”)
14		(0, “y”)
15		(4, “t”)
16		(0, “e”)
17		(8, “s”)
18		(16, “s”)
19		(4, “s”)
20		(16, “a”)
21		(19, “i”)
22		(0, “c”)
23		(0, “k”)
24		(19, “e”)
25		(8, “l”)

Decoding using LZ8

“*sir sid ea*”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9		(1, “t”)
10		(0, “m”)
11		(8, “n”)
12		(7, “a”)

dictionary		token
13		(5, “l”)
14		(0, “y”)
15		(4, “t”)
16		(0, “e”)
17		(8, “s”)
18		(16, “s”)
19		(4, “s”)
20		(16, “a”)
21		(19, “i”)
22		(0, “c”)
23		(0, “k”)
24		(19, “e”)
25		(8, “l”)

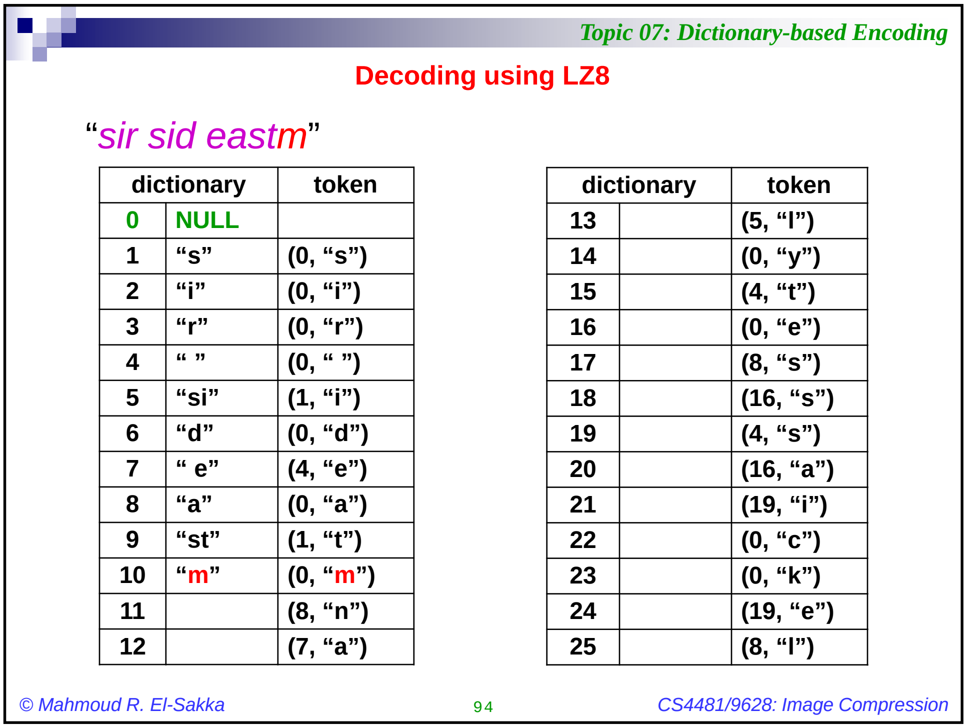


Decoding using LZ8

"*sir sid east*"

dictionary		token
0	NULL	
1	"s"	(0, "s")
2	"i"	(0, "i")
3	"r"	(0, "r")
4	" "	(0, " ")
5	"si"	(1, "i")
6	"d"	(0, "d")
7	" e"	(4, "e")
8	"a"	(0, "a")
9	"st"	(1, "t")
10		(0, "m")
11		(8, "n")
12		(7, "a")

dictionary		token
13		(5, "l")
14		(0, "y")
15		(4, "t")
16		(0, "e")
17		(8, "s")
18		(16, "s")
19		(4, "s")
20		(16, "a")
21		(19, "i")
22		(0, "c")
23		(0, "k")
24		(19, "e")
25		(8, "l")

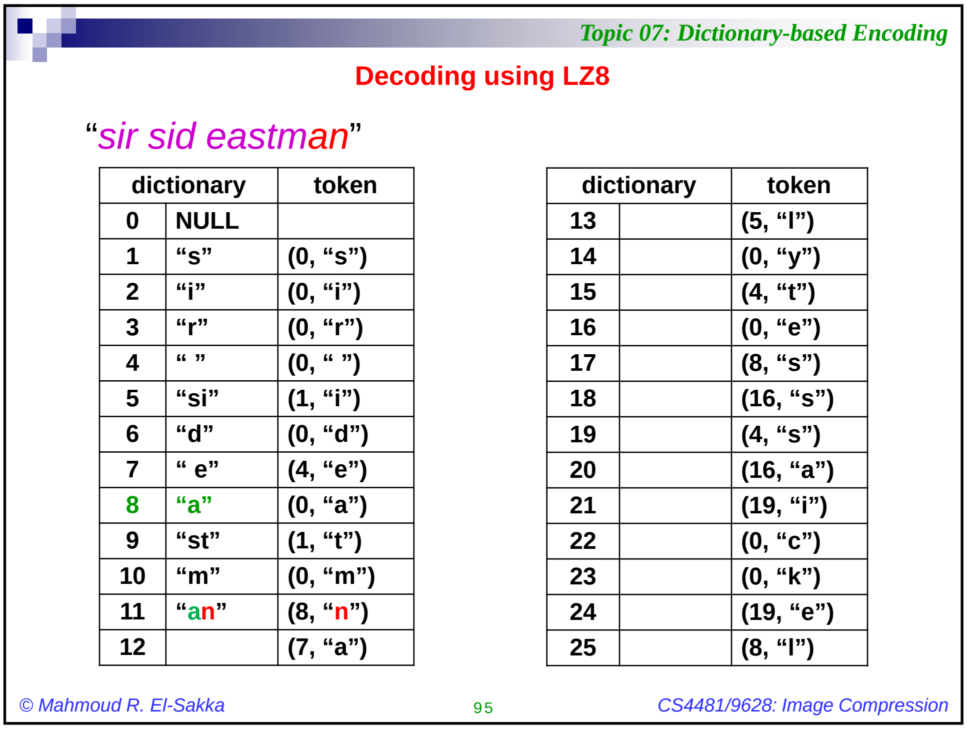


Decoding using LZ8

"*sir sid eastm*"

dictionary		token
0	NULL	
1	"s"	(0, "s")
2	"i"	(0, "i")
3	"r"	(0, "r")
4	" "	(0, " ")
5	"si"	(1, "i")
6	"d"	(0, "d")
7	" e"	(4, "e")
8	"a"	(0, "a")
9	"st"	(1, "t")
10	"m"	(0, "m")
11		(8, "n")
12		(7, "a")

dictionary		token
13		(5, "l")
14		(0, "y")
15		(4, "t")
16		(0, "e")
17		(8, "s")
18		(16, "s")
19		(4, "s")
20		(16, "a")
21		(19, "i")
22		(0, "c")
23		(0, "k")
24		(19, "e")
25		(8, "l")

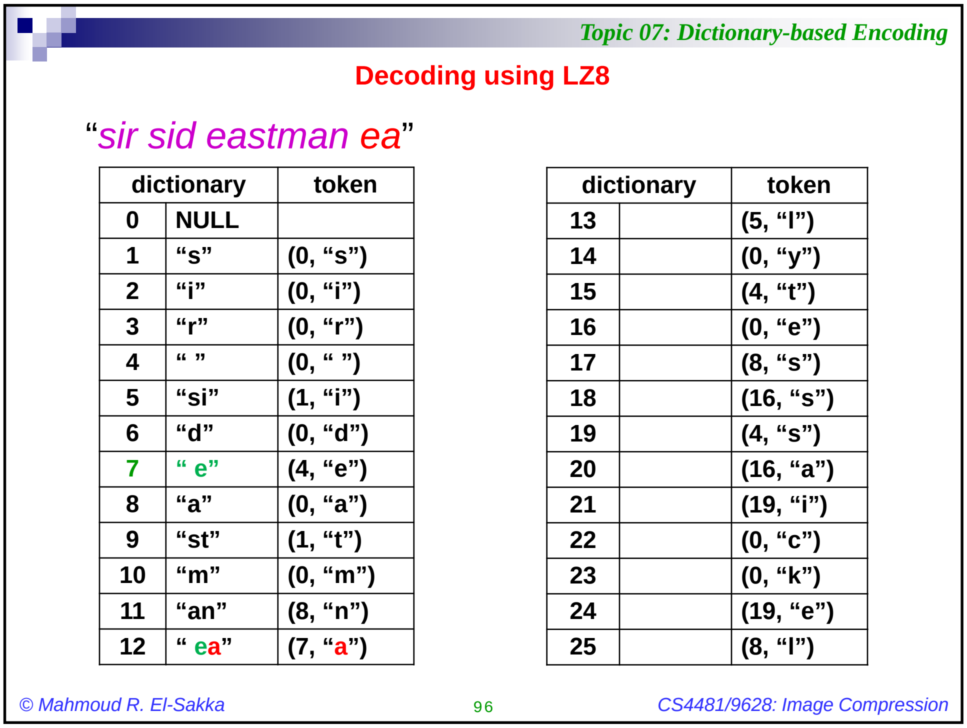


Decoding using LZ8

“*sir sid eastman*”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12		(7, “a”)

dictionary		token
13		(5, “l”)
14		(0, “y”)
15		(4, “t”)
16		(0, “e”)
17		(8, “s”)
18		(16, “s”)
19		(4, “s”)
20		(16, “a”)
21		(19, “i”)
22		(0, “c”)
23		(0, “k”)
24		(19, “e”)
25		(8, “l”)



Decoding using LZ8

“*sir sid eastman ea*”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

dictionary		token
13		(5, “l”)
14		(0, “y”)
15		(4, “t”)
16		(0, “e”)
17		(8, “s”)
18		(16, “s”)
19		(4, “s”)
20		(16, “a”)
21		(19, “i”)
22		(0, “c”)
23		(0, “k”)
24		(19, “e”)
25		(8, “l”)

Decoding using LZ8

“*sir sid eastman easil*”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

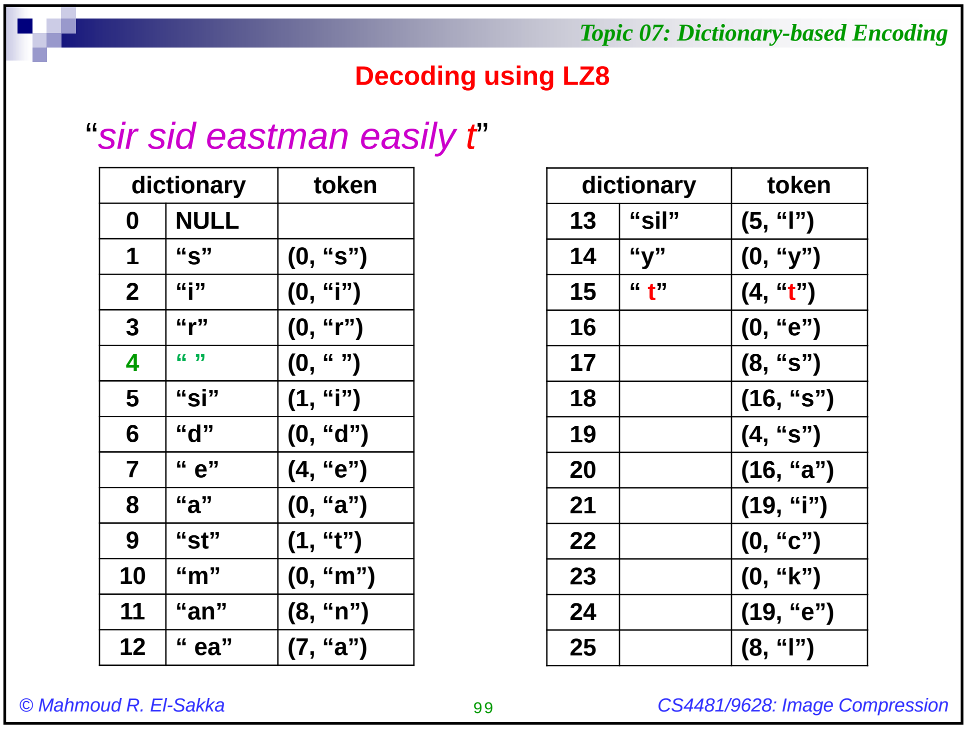
dictionary		token
13	“sil”	(5, “l”)
14		(0, “y”)
15		(4, “t”)
16		(0, “e”)
17		(8, “s”)
18		(16, “s”)
19		(4, “s”)
20		(16, “a”)
21		(19, “i”)
22		(0, “c”)
23		(0, “k”)
24		(19, “e”)
25		(8, “l”)

Decoding using LZ8

“*sir sid eastman easily*”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

dictionary		token
13	“sil”	(5, “l”)
14	“y”	(0, “y”)
15		(4, “t”)
16		(0, “e”)
17		(8, “s”)
18		(16, “s”)
19		(4, “s”)
20		(16, “a”)
21		(19, “i”)
22		(0, “c”)
23		(0, “k”)
24		(19, “e”)
25		(8, “l”)

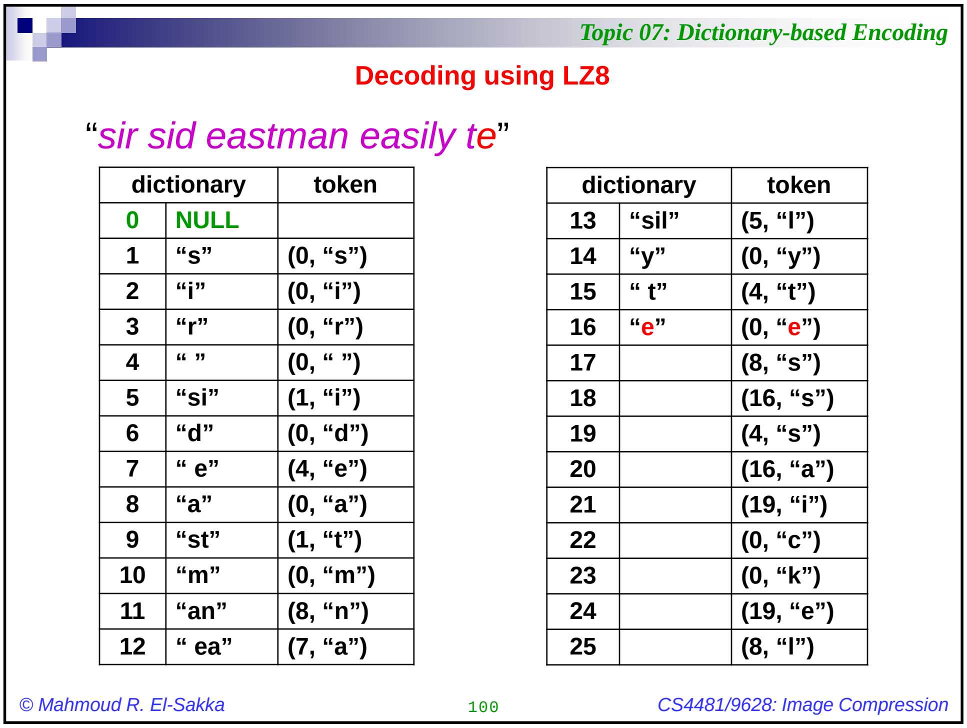


Decoding using LZ8

“*sir sid eastman easily t*”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

dictionary		token
13	“sil”	(5, “l”)
14	“y”	(0, “y”)
15	“ t”	(4, “t”)
16		(0, “e”)
17		(8, “s”)
18		(16, “s”)
19		(4, “s”)
20		(16, “a”)
21		(19, “i”)
22		(0, “c”)
23		(0, “k”)
24		(19, “e”)
25		(8, “l”)

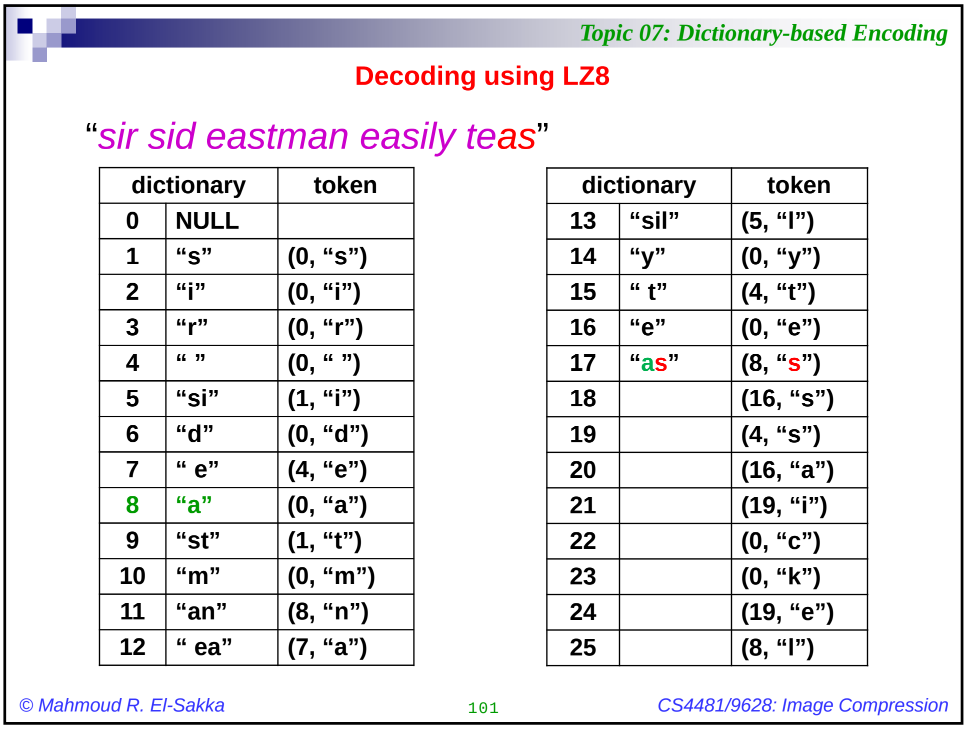


Decoding using LZ8

“*sir sid eastman easily te*”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

dictionary		token
13	“sil”	(5, “l”)
14	“y”	(0, “y”)
15	“ t”	(4, “t”)
16	“e”	(0, “e”)
17		(8, “s”)
18		(16, “s”)
19		(4, “s”)
20		(16, “a”)
21		(19, “i”)
22		(0, “c”)
23		(0, “k”)
24		(19, “e”)
25		(8, “l”)

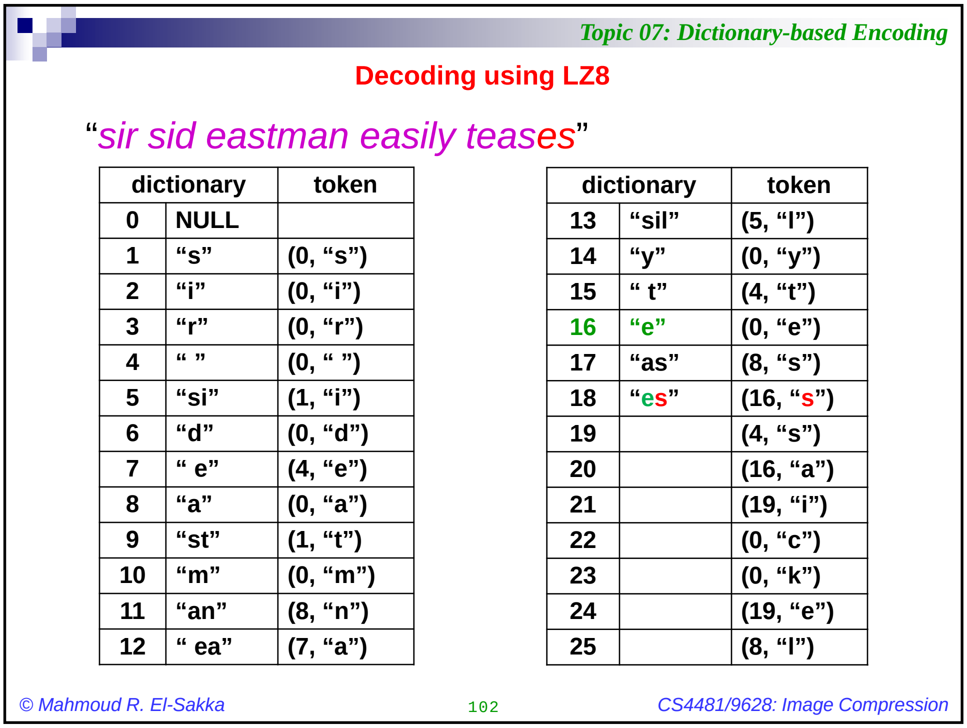


Decoding using LZ8

“*sir sid eastman easily teas*”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

dictionary		token
13	“sil”	(5, “l”)
14	“y”	(0, “y”)
15	“ t”	(4, “t”)
16	“e”	(0, “e”)
17	“as”	(8, “s”)
18		(16, “s”)
19		(4, “s”)
20		(16, “a”)
21		(19, “i”)
22		(0, “c”)
23		(0, “k”)
24		(19, “e”)
25		(8, “l”)

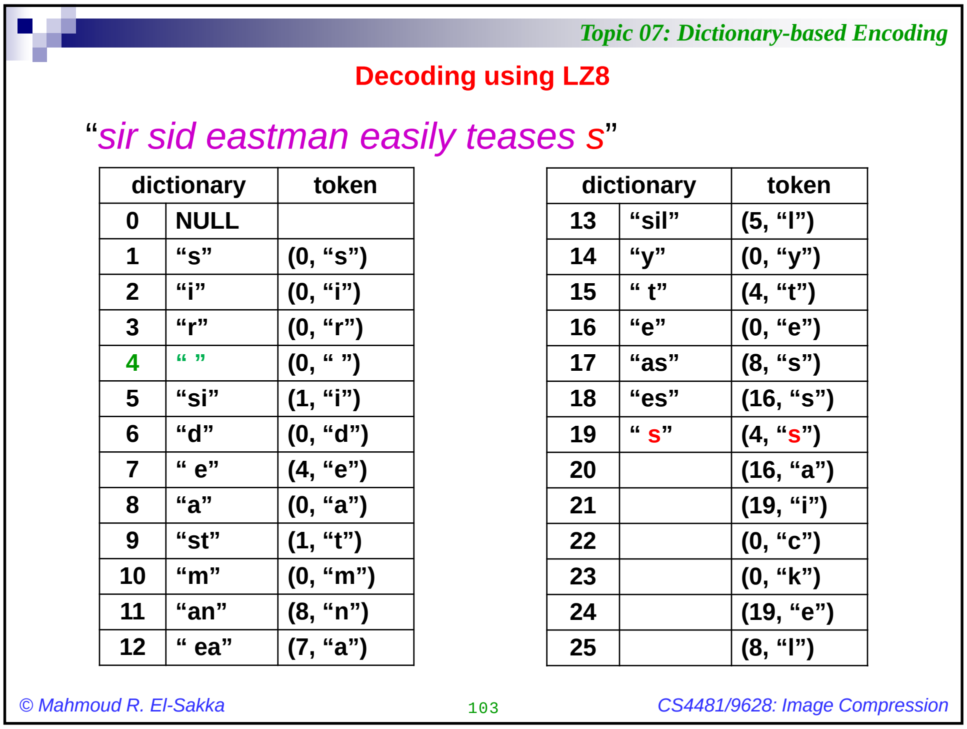


Decoding using LZ8

“*sir sid eastman easily teases*”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

dictionary		token
13	“sil”	(5, “l”)
14	“y”	(0, “y”)
15	“ t”	(4, “t”)
16	“e”	(0, “e”)
17	“as”	(8, “s”)
18	“es”	(16, “s”)
19		(4, “s”)
20		(16, “a”)
21		(19, “i”)
22		(0, “c”)
23		(0, “k”)
24		(19, “e”)
25		(8, “l”)

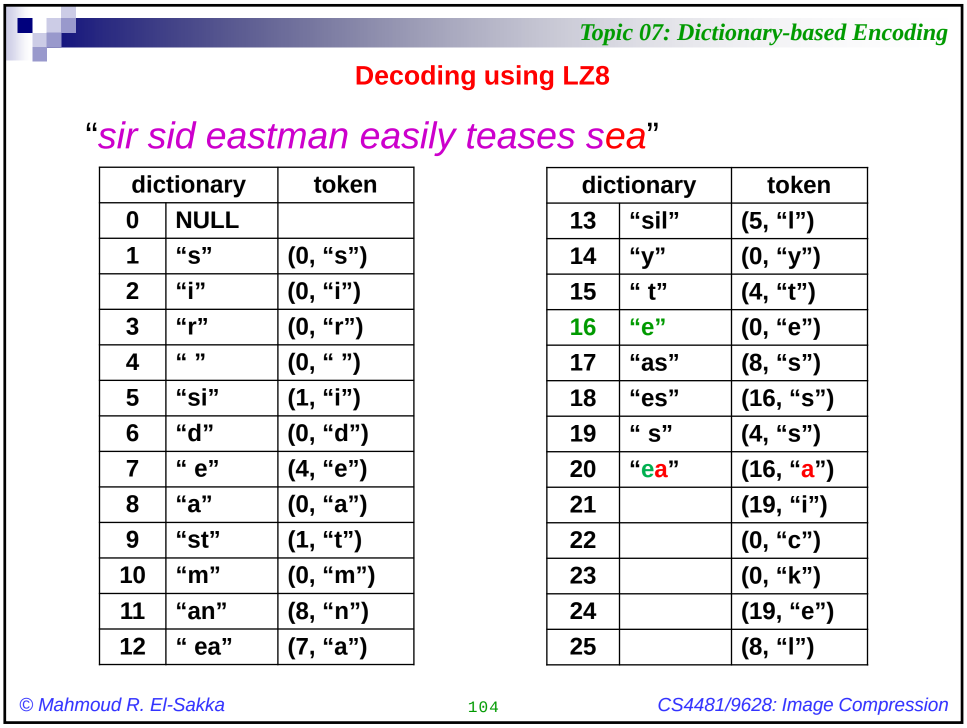


Decoding using LZ8

“sir sid eastman easily teases s”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

dictionary		token
13	“sil”	(5, “l”)
14	“y”	(0, “y”)
15	“ t”	(4, “t”)
16	“e”	(0, “e”)
17	“as”	(8, “s”)
18	“es”	(16, “s”)
19	“ s”	(4, “s”)
20		(16, “a”)
21		(19, “i”)
22		(0, “c”)
23		(0, “k”)
24		(19, “e”)
25		(8, “l”)

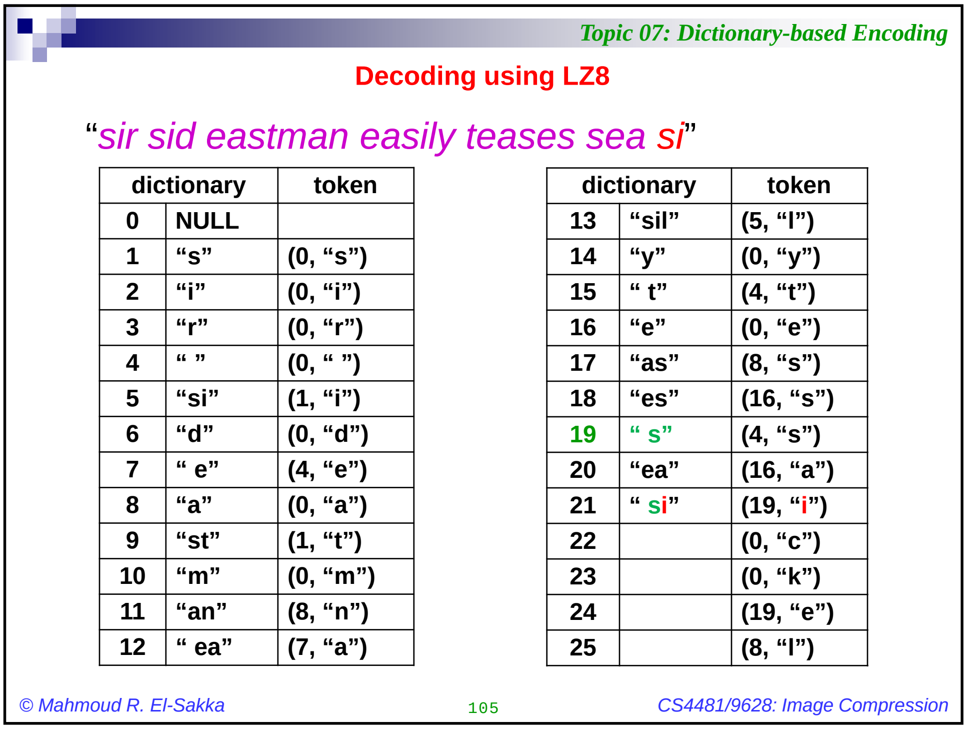


Decoding using LZ8

“sir sid eastman easily teases sea”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

dictionary		token
13	“sil”	(5, “l”)
14	“y”	(0, “y”)
15	“ t”	(4, “t”)
16	“e”	(0, “e”)
17	“as”	(8, “s”)
18	“es”	(16, “s”)
19	“ s”	(4, “s”)
20	“ea”	(16, “a”)
21		(19, “i”)
22		(0, “c”)
23		(0, “k”)
24		(19, “e”)
25		(8, “l”)

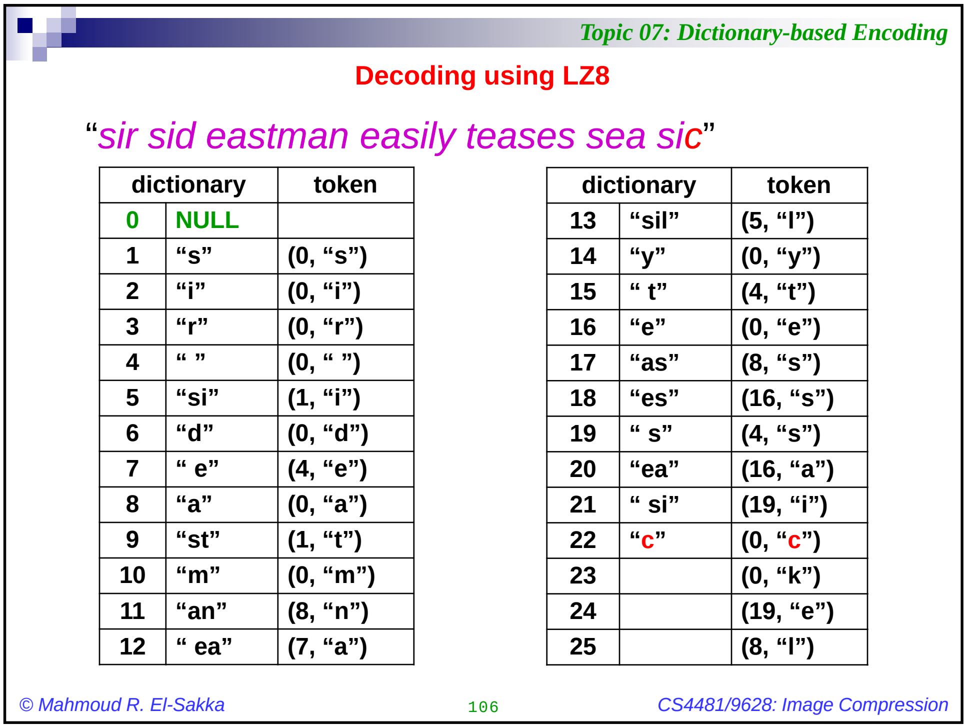


Decoding using LZ8

“*sir sid eastman easily teases sea si*”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

dictionary		token
13	“sil”	(5, “l”)
14	“y”	(0, “y”)
15	“ t”	(4, “t”)
16	“e”	(0, “e”)
17	“as”	(8, “s”)
18	“es”	(16, “s”)
19	“ s”	(4, “s”)
20	“ea”	(16, “a”)
21	“ si”	(19, “i”)
22		(0, “c”)
23		(0, “k”)
24		(19, “e”)
25		(8, “l”)



Decoding using LZ8

“*sir sid eastman easily teases sea sic*”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

dictionary		token
13	“sil”	(5, “l”)
14	“y”	(0, “y”)
15	“ t”	(4, “t”)
16	“e”	(0, “e”)
17	“as”	(8, “s”)
18	“es”	(16, “s”)
19	“ s”	(4, “s”)
20	“ea”	(16, “a”)
21	“ si”	(19, “i”)
22	“c”	(0, “c”)
23		(0, “k”)
24		(19, “e”)
25		(8, “l”)

Decoding using LZ8

“sir sid eastman easily teases sea sick”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

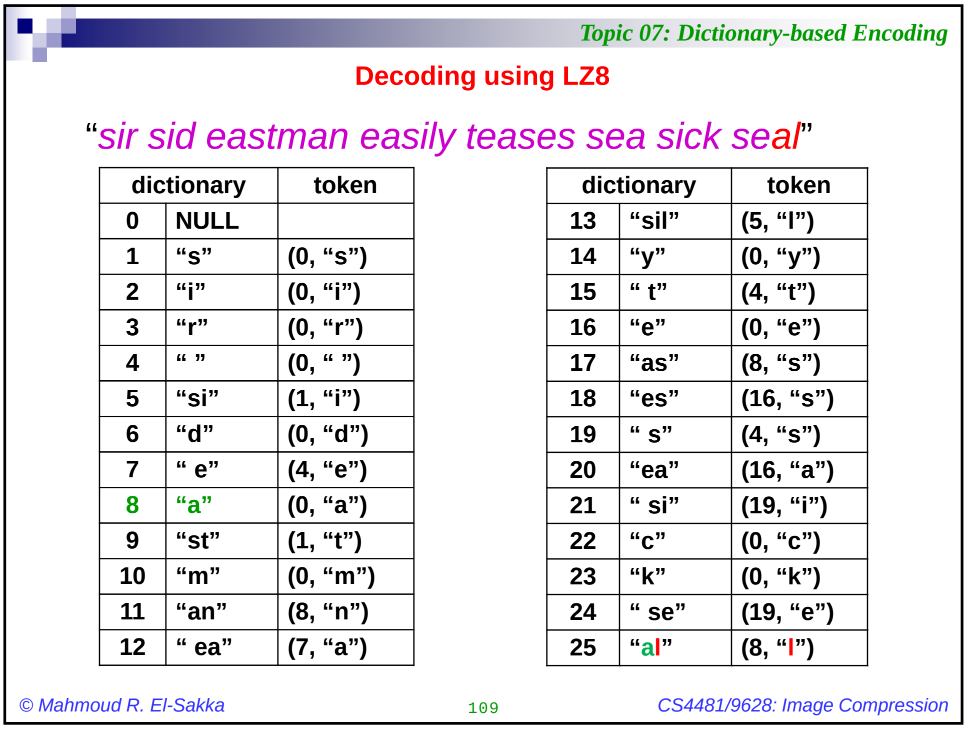
dictionary		token
13	“sil”	(5, “l”)
14	“y”	(0, “y”)
15	“ t”	(4, “t”)
16	“e”	(0, “e”)
17	“as”	(8, “s”)
18	“es”	(16, “s”)
19	“ s”	(4, “s”)
20	“ea”	(16, “a”)
21	“ si”	(19, “i”)
22	“c”	(0, “c”)
23	“k”	(0, “k”)
24		(19, “e”)
25		(8, “l”)

Decoding using LZ8

“sir sid eastman easily teases sea sick se”

dictionary		token
0	NULL	
1	“s”	(0, “s”)
2	“i”	(0, “i”)
3	“r”	(0, “r”)
4	“ ”	(0, “ ”)
5	“si”	(1, “i”)
6	“d”	(0, “d”)
7	“ e”	(4, “e”)
8	“a”	(0, “a”)
9	“st”	(1, “t”)
10	“m”	(0, “m”)
11	“an”	(8, “n”)
12	“ ea”	(7, “a”)

dictionary		token
13	“sil”	(5, “l”)
14	“y”	(0, “y”)
15	“ t”	(4, “t”)
16	“e”	(0, “e”)
17	“as”	(8, “s”)
18	“es”	(16, “s”)
19	“ s”	(4, “s”)
20	“ea”	(16, “a”)
21	“ si”	(19, “i”)
22	“c”	(0, “c”)
23	“k”	(0, “k”)
24	“ se”	(19, “e”)
25		(8, “l”)

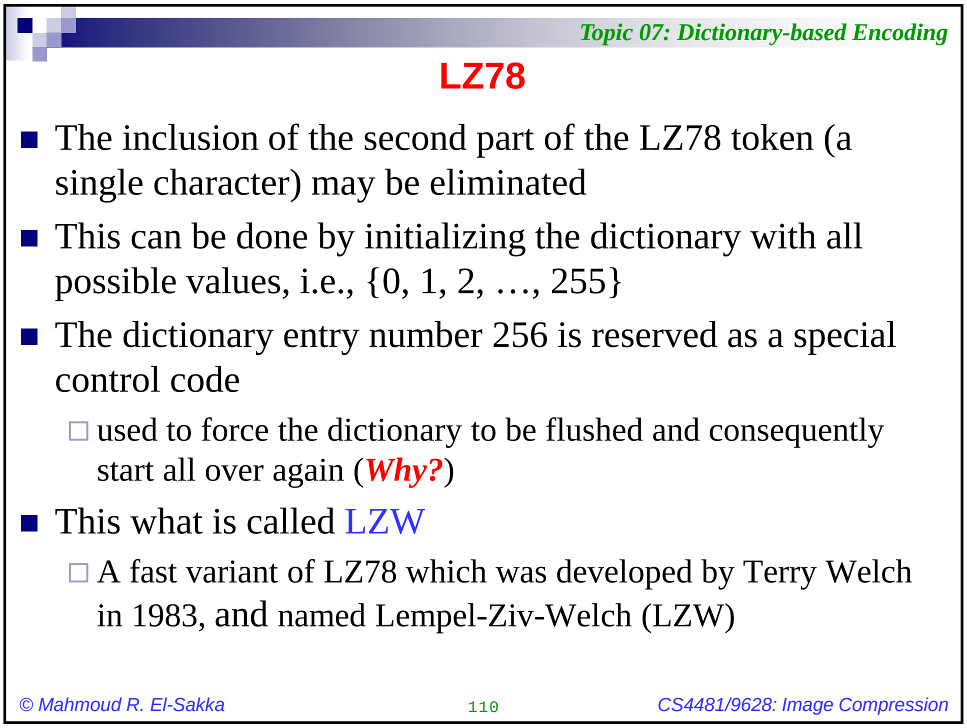


Decoding using LZ8

"sir sid eastman easily teases sea sick seal"

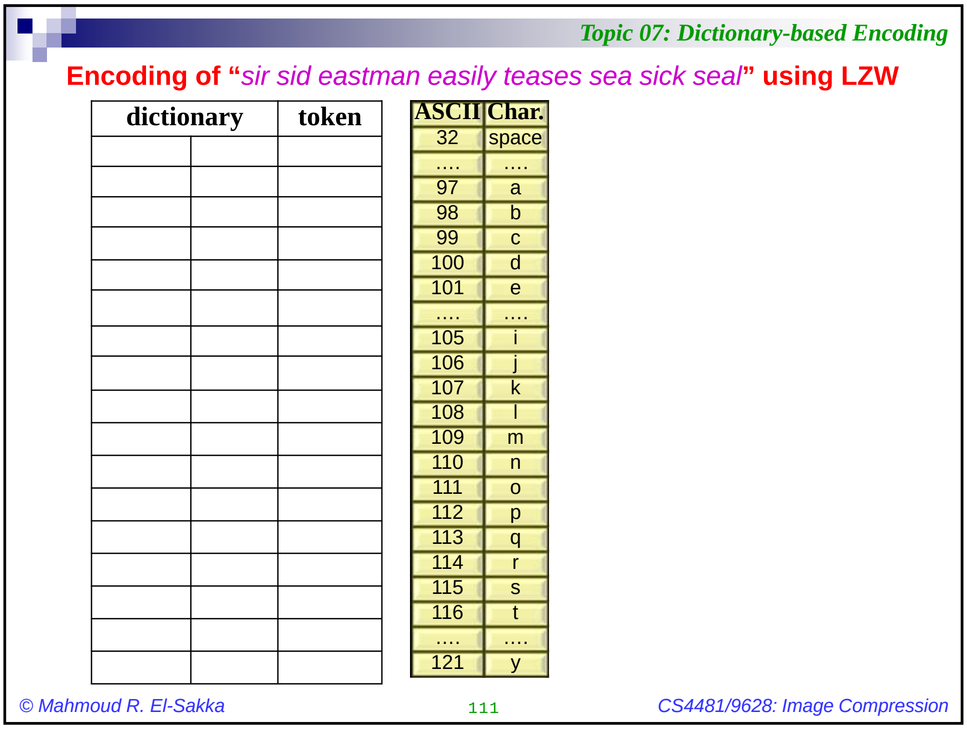
dictionary		token
0	NULL	
1	"s"	(0, "s")
2	"i"	(0, "i")
3	"r"	(0, "r")
4	" "	(0, " ")
5	"si"	(1, "i")
6	"d"	(0, "d")
7	" e"	(4, "e")
8	"a"	(0, "a")
9	"st"	(1, "t")
10	"m"	(0, "m")
11	"an"	(8, "n")
12	" ea"	(7, "a")

dictionary		token
13	"sil"	(5, "l")
14	"y"	(0, "y")
15	" t"	(4, "t")
16	"e"	(0, "e")
17	"as"	(8, "s")
18	"es"	(16, "s")
19	" s"	(4, "s")
20	"ea"	(16, "a")
21	" si"	(19, "i")
22	"c"	(0, "c")
23	"k"	(0, "k")
24	" se"	(19, "e")
25	"al"	(8, "l")



LZ78

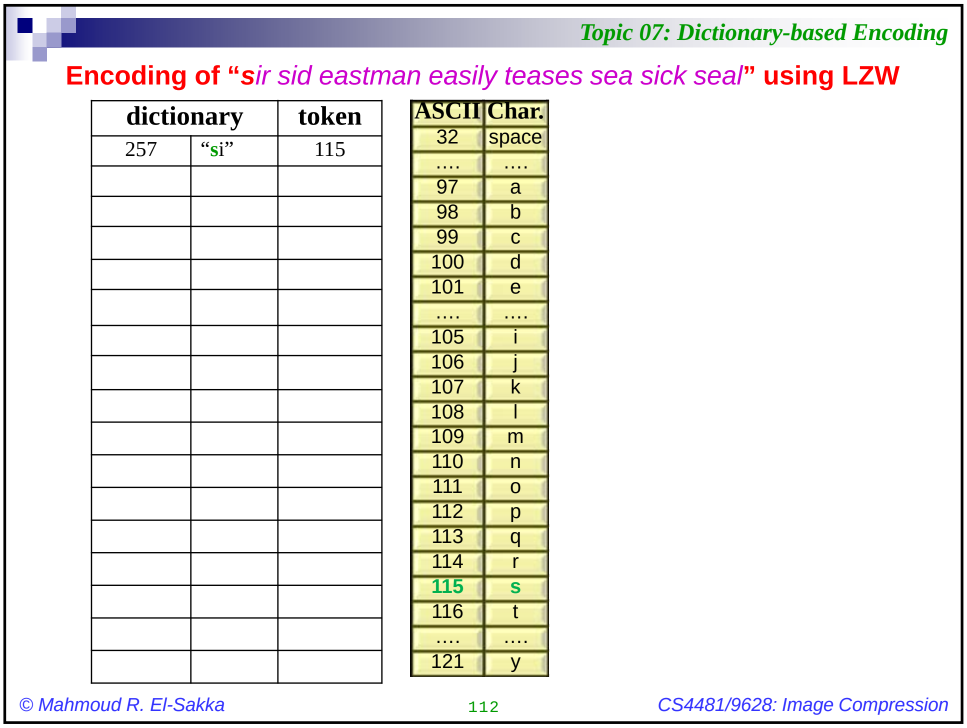
- The inclusion of the second part of the LZ78 token (a single character) may be eliminated
- This can be done by initializing the dictionary with all possible values, i.e., {0, 1, 2, …, 255}
- The dictionary entry number 256 is reserved as a special control code
  - □ used to force the dictionary to be flushed and consequently start all over again (***Why?***)
- This what is called LZW
  - □ A fast variant of LZ78 which was developed by Terry Welch in 1983, and named Lempel-Ziv-Welch (LZW)



Encoding of "*sir sid eastman easily teases sea sick seal*" using LZW

dictionary		token

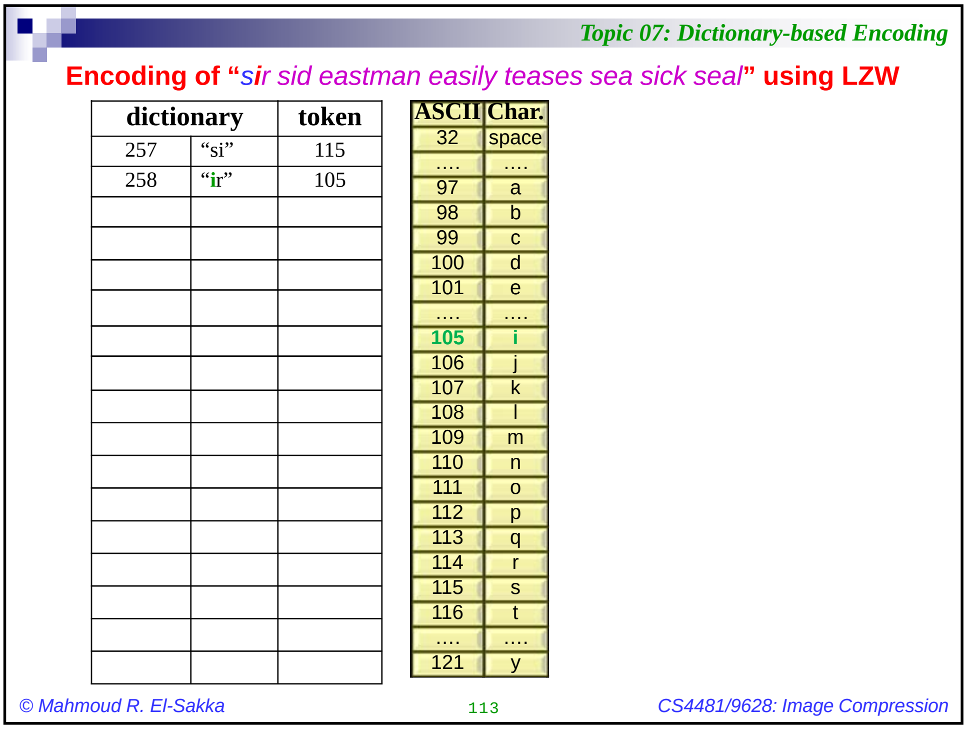
ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y



Encoding of "*sir sid eastman easily teases sea sick seal*" using LZW

dictionary		token
257	"si"	115

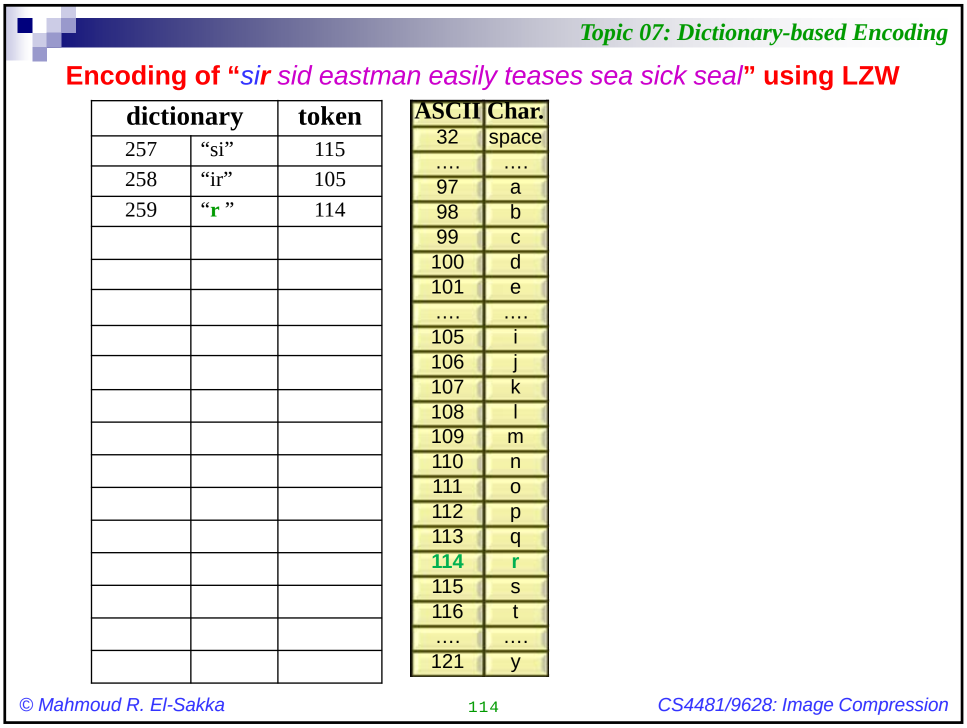
ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y



Encoding of "sir sid eastman easily teases sea sick seal" using LZW

dictionary		token
257	"si"	115
258	"ir"	105

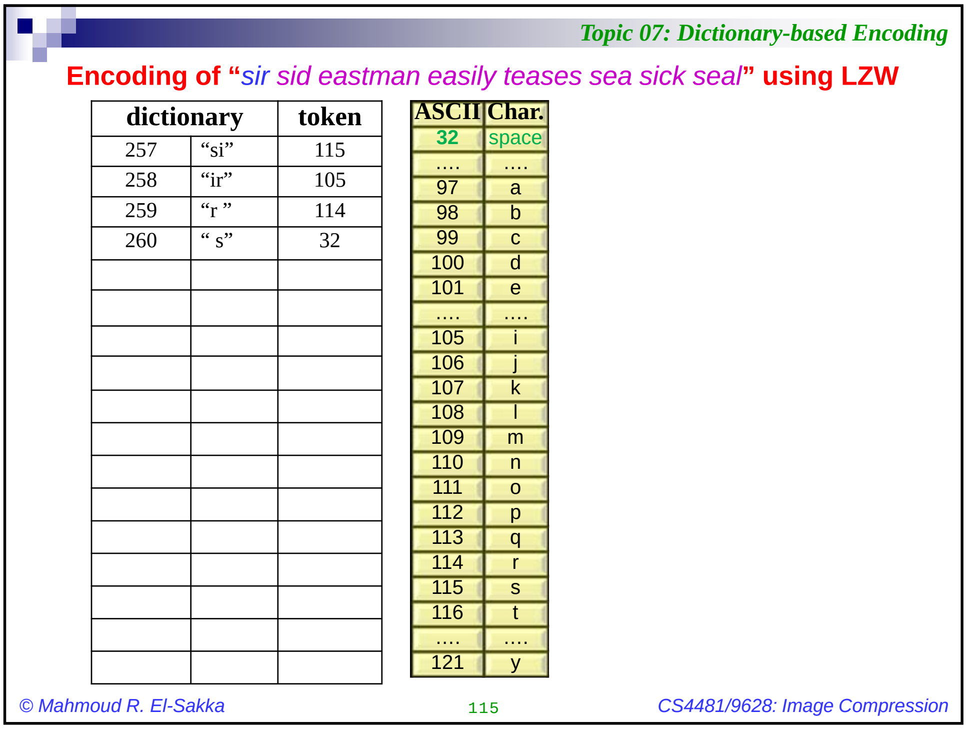
ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y



Encoding of "sir sid eastman easily teases sea sick seal" using LZW

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114

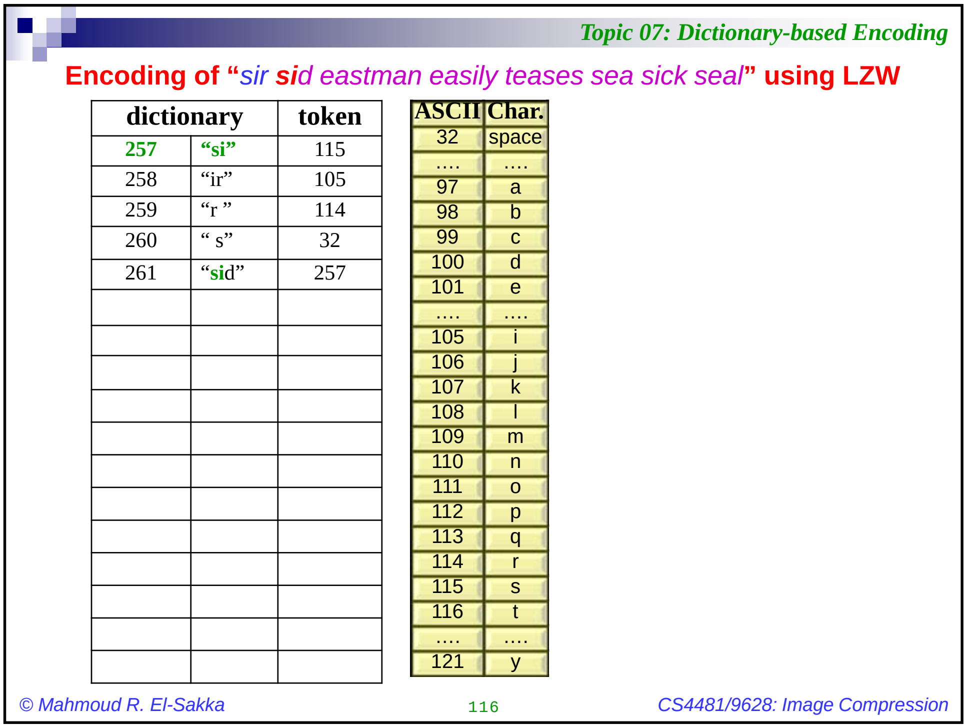
ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y



Encoding of "sir sid eastman easily teases sea sick seal" using LZW

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32

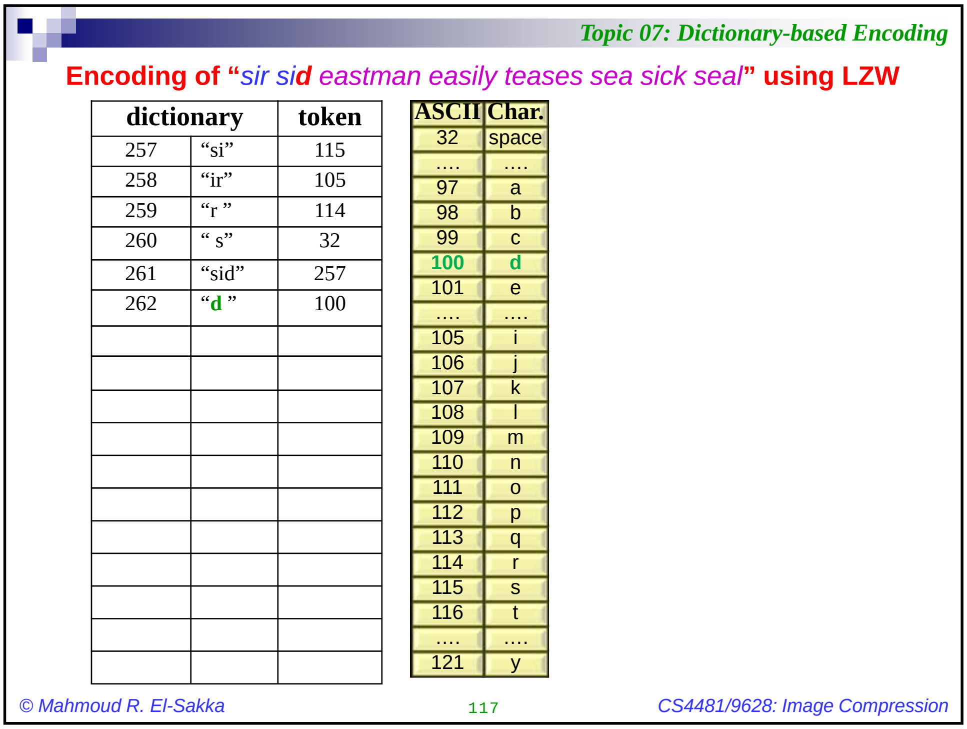
ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y



Encoding of "sir sid eastman easily teases sea sick seal" using LZW

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257

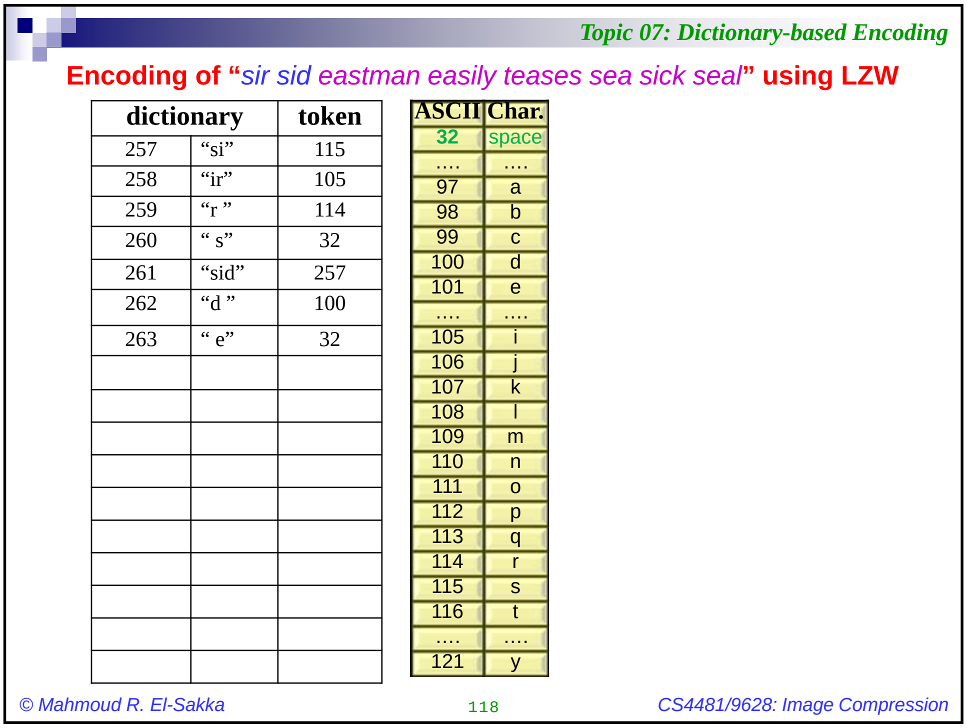
ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y



Encoding of “*sir si**d** eastman easily teases sea sick seal*” using LZW

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“**d** ”	100

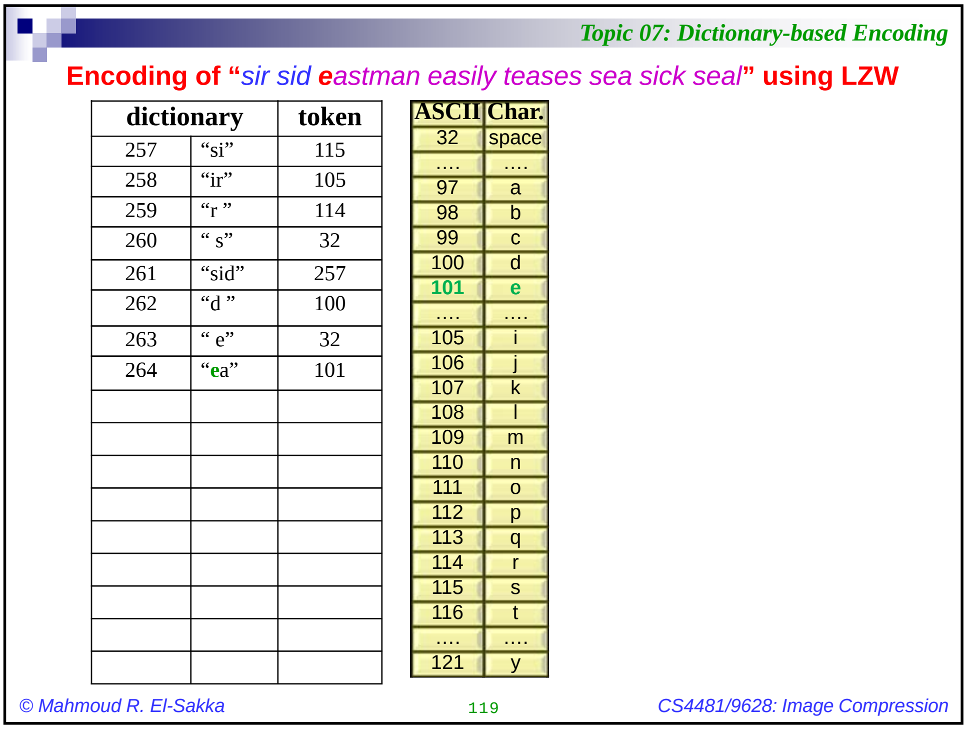
ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	**d**
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y



Encoding of “*sir sid eastman easily teases sea sick seal*” using LZW

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32

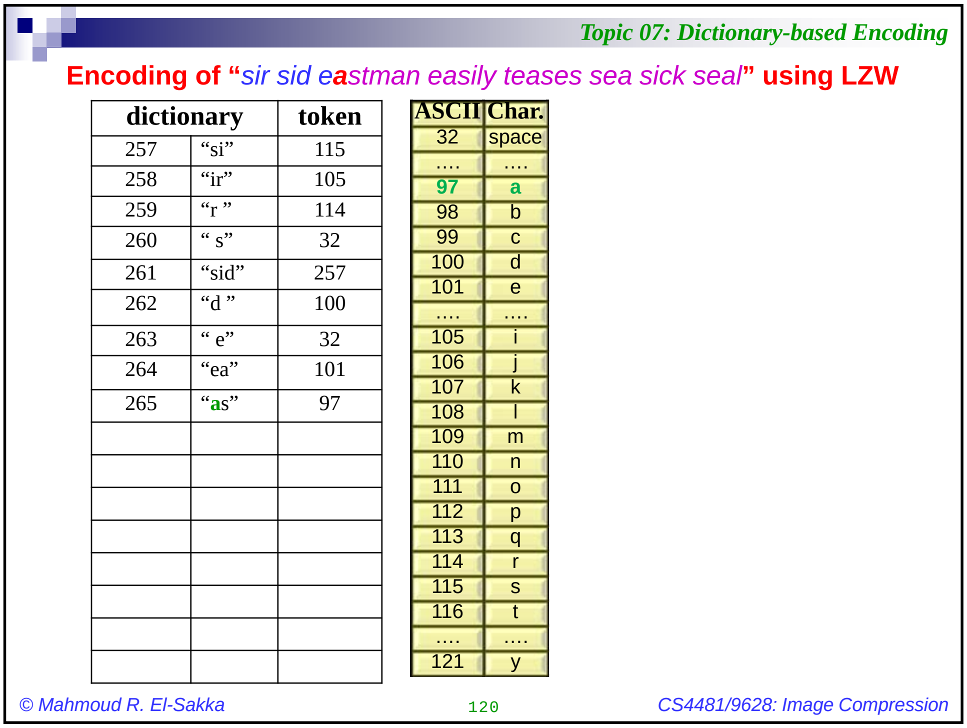
ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y



Encoding of “sir sid eastman easily teases sea sick seal” using LZW

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101

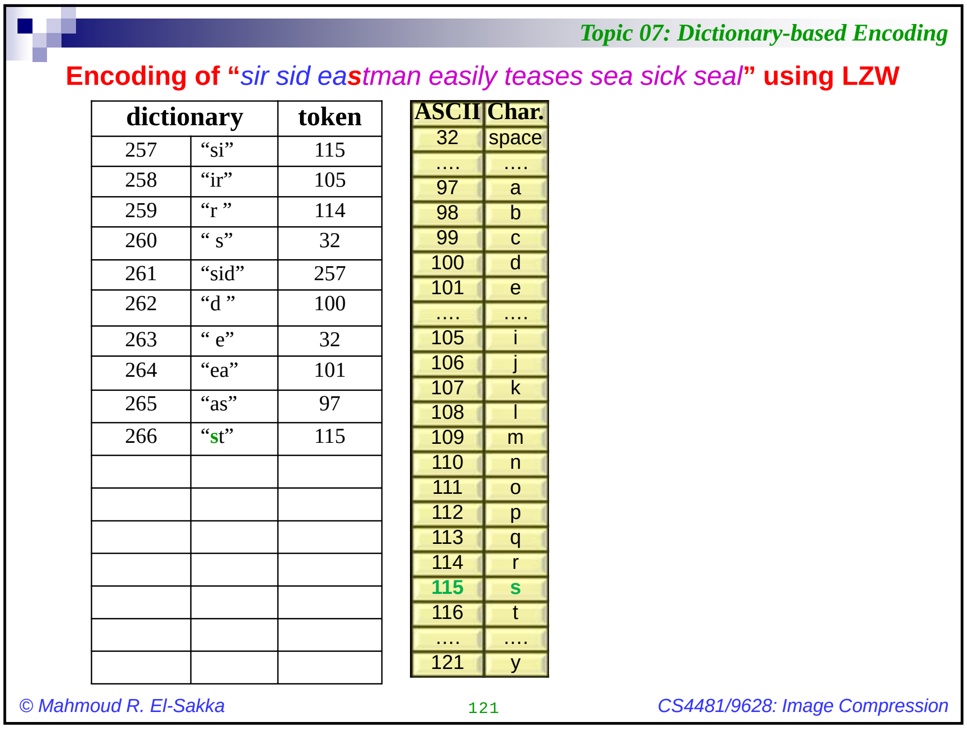
ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y



Encoding of “sir sid eastman easily teases sea sick seal” using LZW

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97

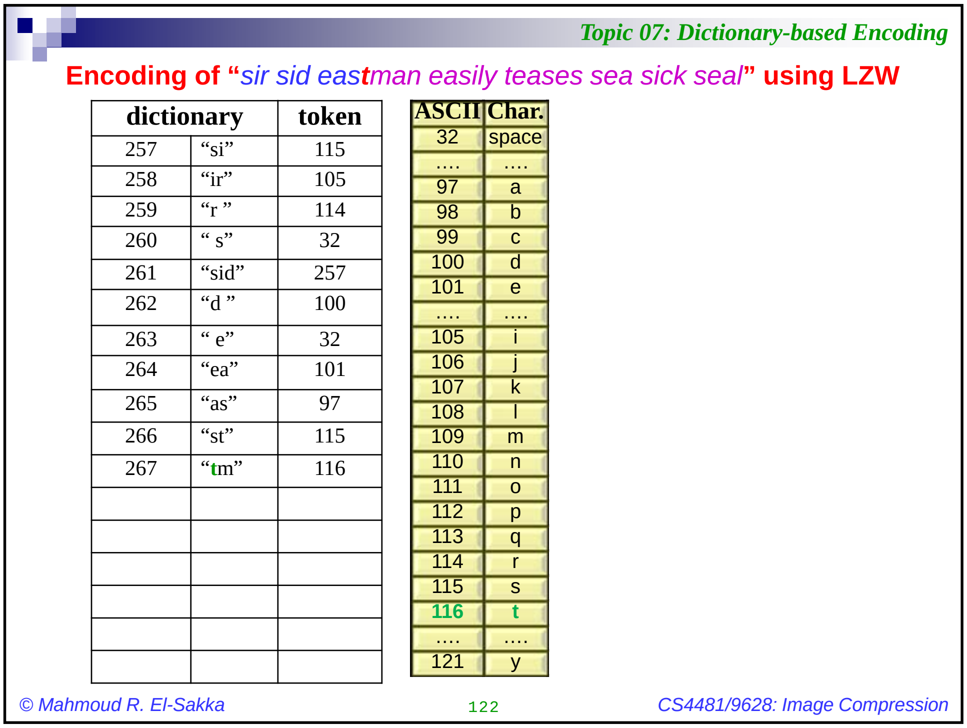
ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y



Encoding of “sir sid eastman easily teases sea sick seal” using LZW

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115

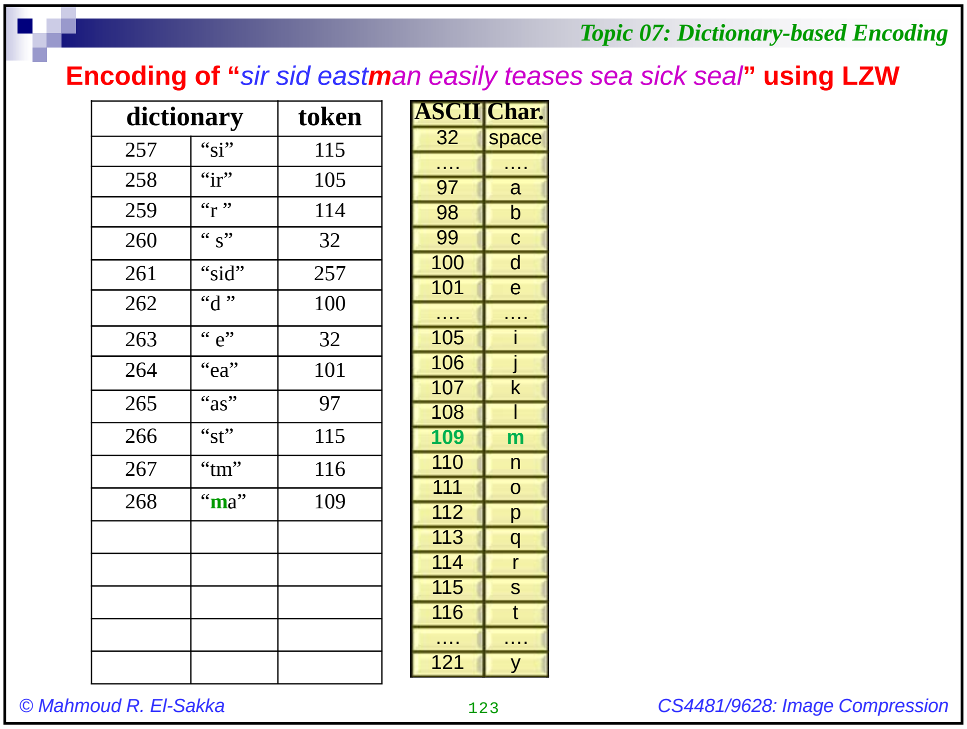
ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y



Encoding of “sir sid eastman easily teases sea sick seal” using LZW

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116

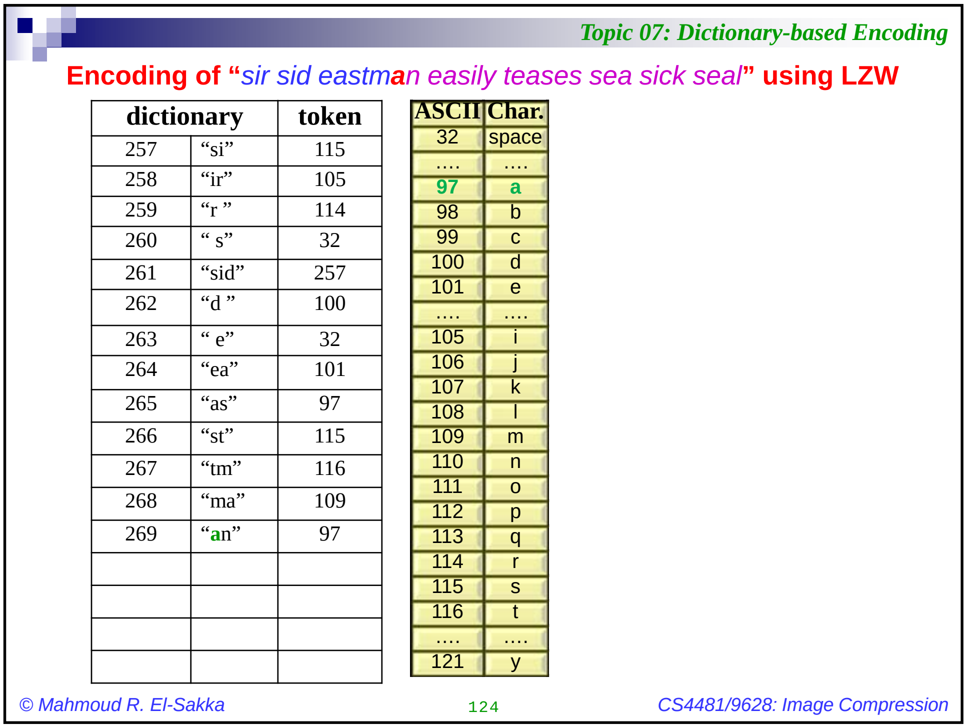
ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y



Encoding of “sir sid east**m**an easily teases sea sick seal” using LZW

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“**m**a”	109

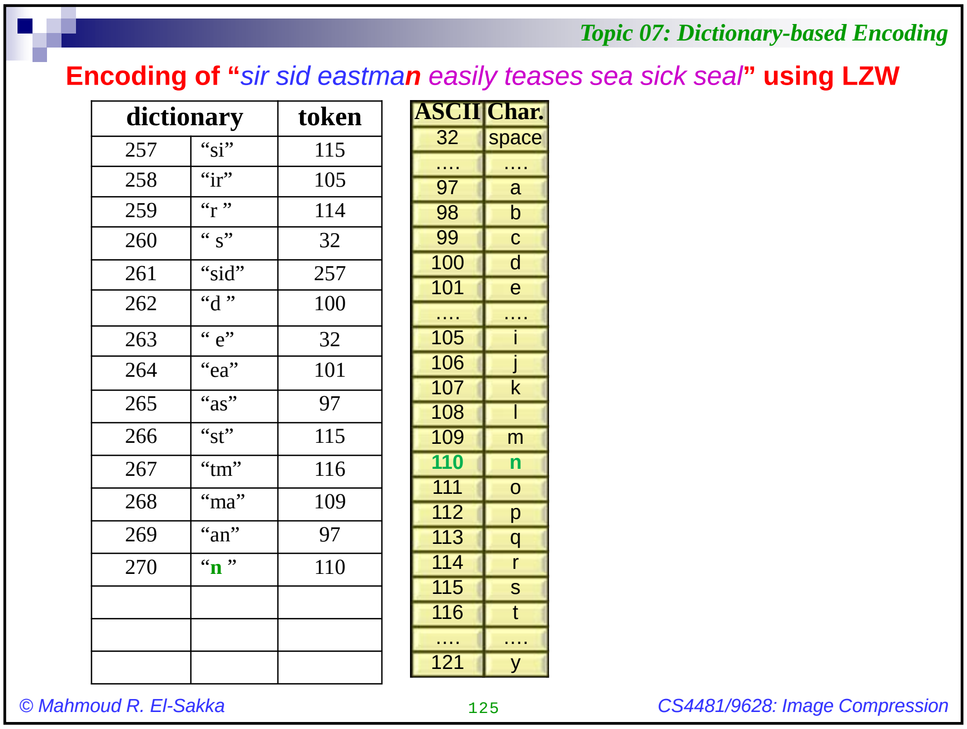
ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	**m**
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y



Encoding of “sir sid eastm**a**n easily teases sea sick seal” using LZW

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“**a**n”	97

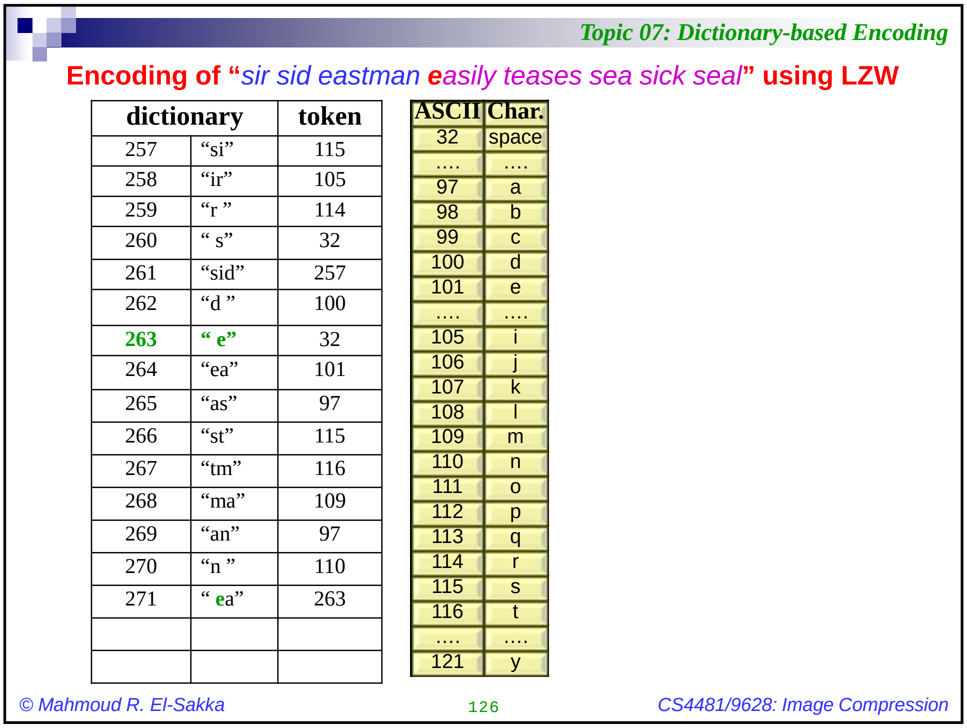
ASCII	Char.
32	space
….	….
97	**a**
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y



Encoding of “sir sid eastma**n** easily teases sea sick seal” using LZW

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“**n** ”	110

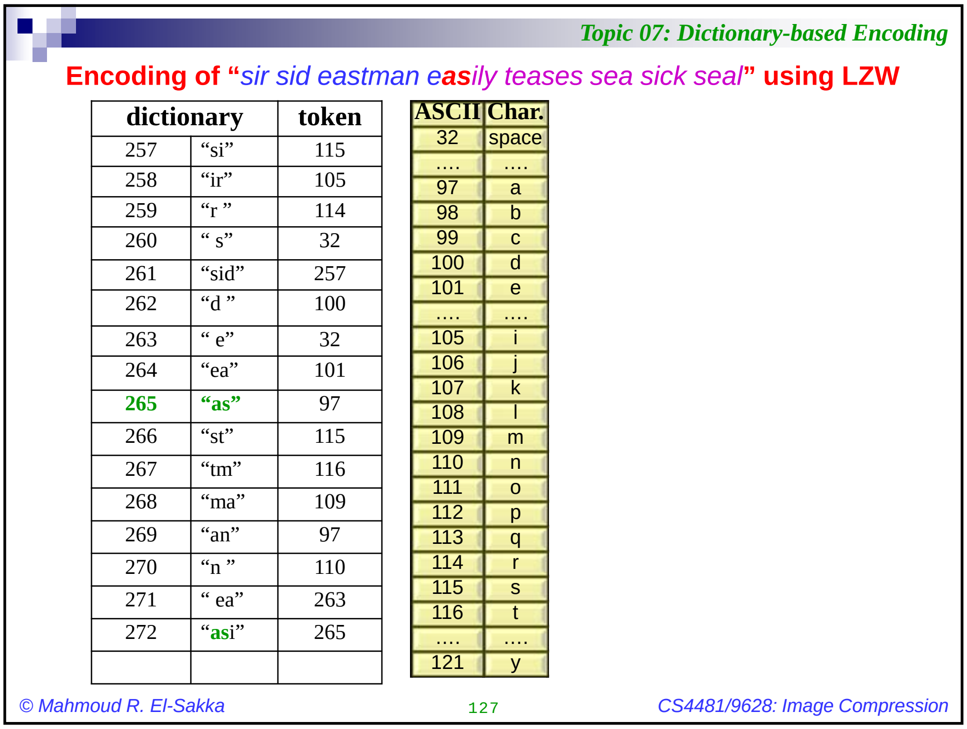
ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	**n**
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y



Encoding of “sir sid eastman **e**asily teases sea sick seal” using LZW

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	**“ e”**	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ **ea**”	263

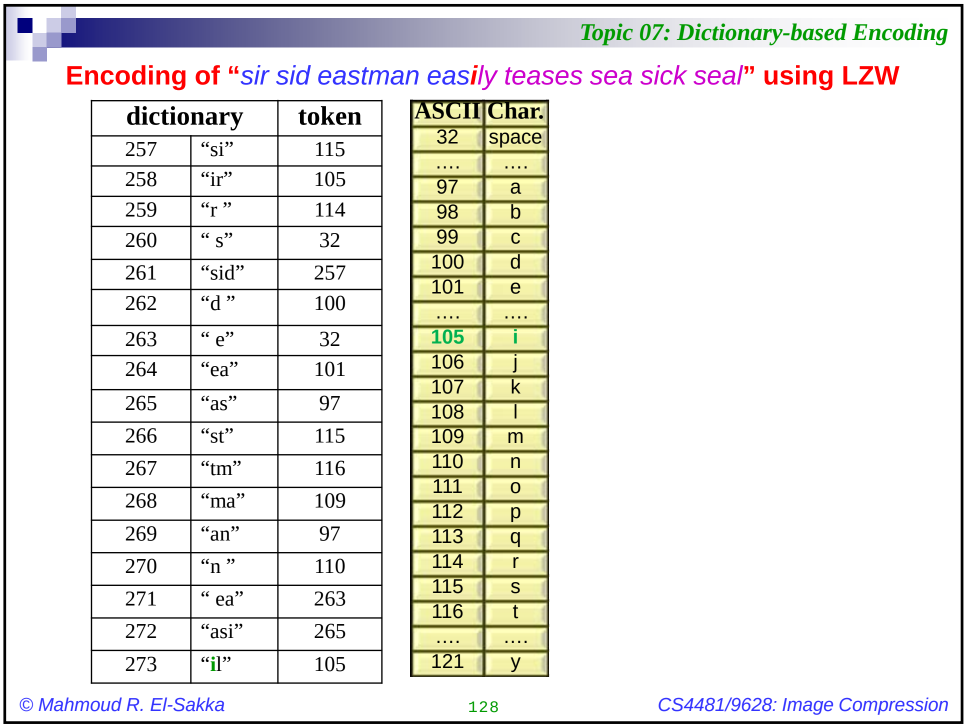
ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y



Encoding of “sir sid eastman e**as**ily teases sea sick seal” using LZW

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	**“as”**	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“**as**i”	265

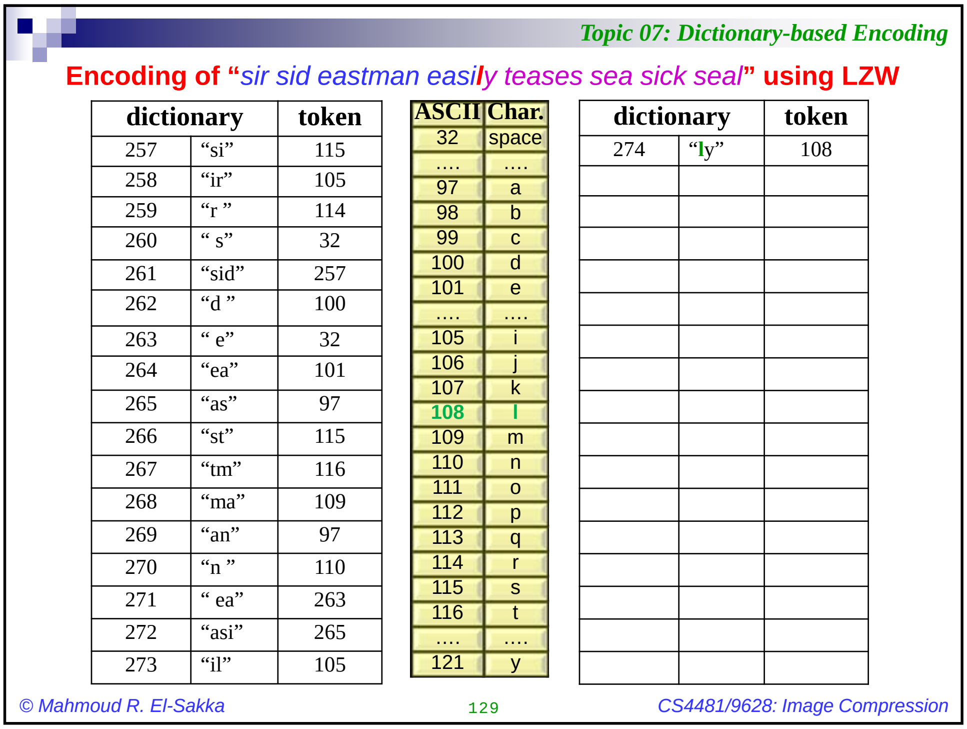
ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y



Encoding of “sir sid eastman eas**i**ly teases sea sick seal” using LZW

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“asi”	265
273	“**i**l”	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	**i**
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

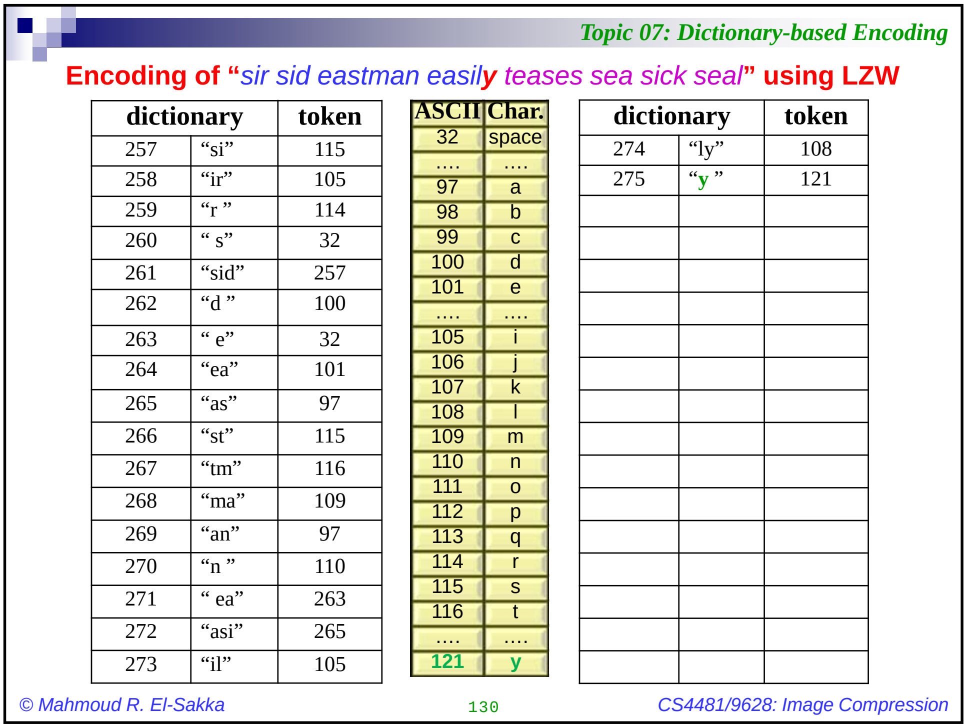


Encoding of “*sir sid eastman easily teases sea sick seal*” using LZW

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“asi”	265
273	“il”	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	“ly”	108

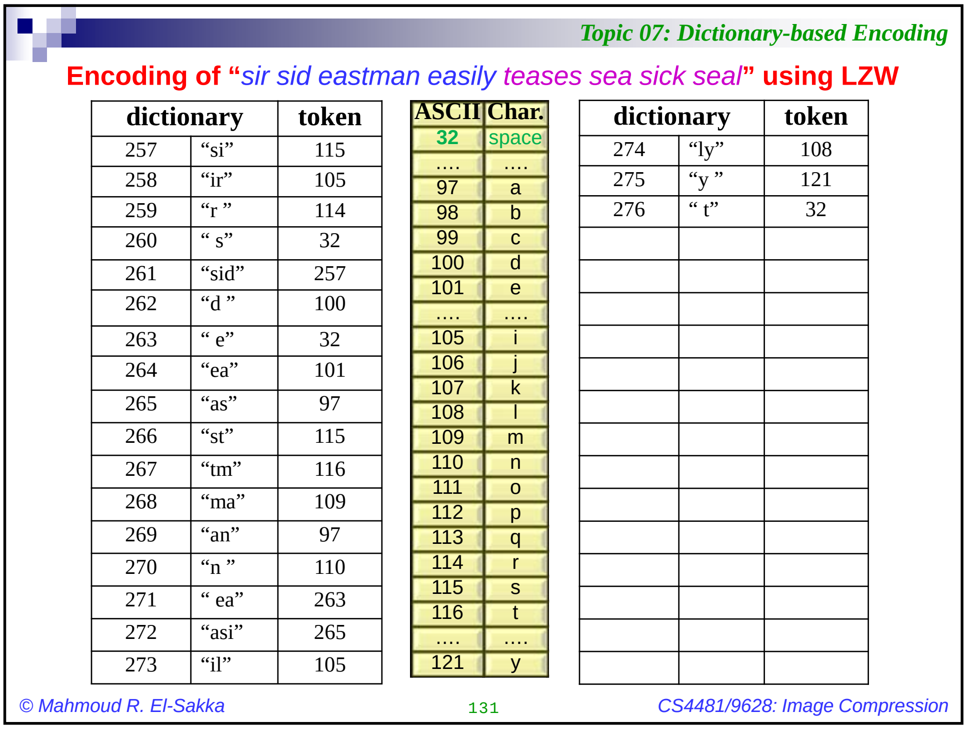


Encoding of “*sir sid eastman easily teases sea sick seal*” using LZW

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“asi”	265
273	“il”	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	“ly”	108
275	“y ”	121



Topic 07: Dictionary-based Encoding

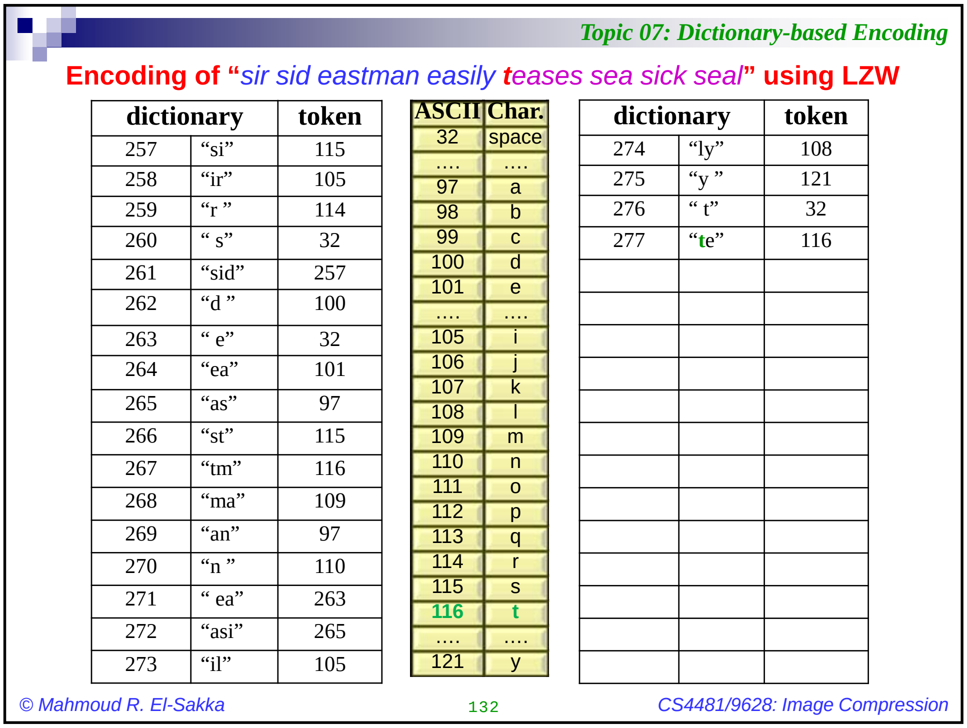
Encoding of "*sir sid eastman easily teases sea sick seal*" using LZW

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	"ea"	101
265	"as"	97
266	"st"	115
267	"tm"	116
268	"ma"	109
269	"an"	97
270	"n "	110
271	" ea"	263
272	"asi"	265
273	"il"	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	"ly"	108
275	"y "	121
276	" t"	32

© Mahmoud R. El-Sakka 131 CS4481/9628: Image Compression



Topic 07: Dictionary-based Encoding

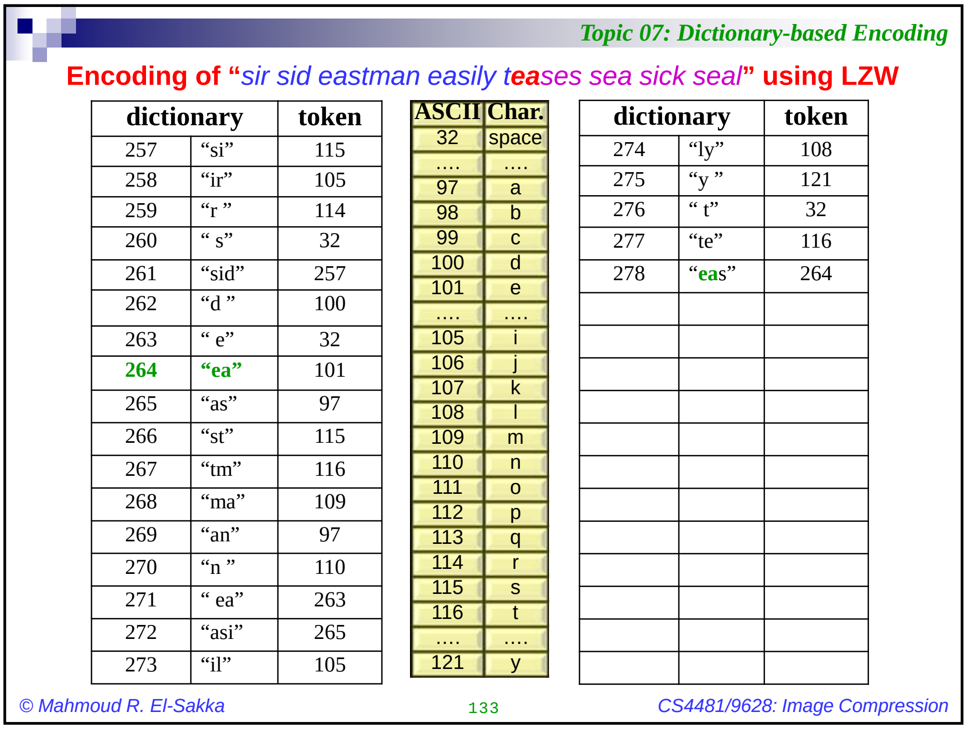
Encoding of "*sir sid eastman easily teases sea sick seal*" using LZW

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	"ea"	101
265	"as"	97
266	"st"	115
267	"tm"	116
268	"ma"	109
269	"an"	97
270	"n "	110
271	" ea"	263
272	"asi"	265
273	"il"	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	"ly"	108
275	"y "	121
276	" t"	32
277	"te"	116

© Mahmoud R. El-Sakka 132 CS4481/9628: Image Compression

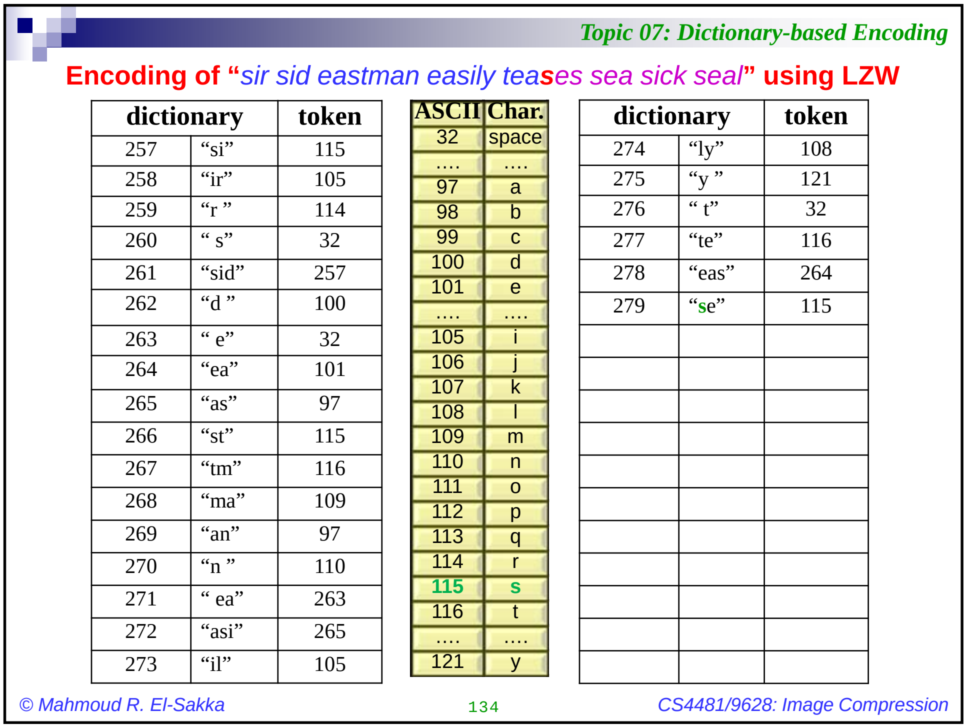


Encoding of "*sir sid eastman easily teases sea sick seal*" using LZW

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	**"ea"**	101
265	"as"	97
266	"st"	115
267	"tm"	116
268	"ma"	109
269	"an"	97
270	"n "	110
271	" ea"	263
272	"asi"	265
273	"il"	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	"ly"	108
275	"y "	121
276	" t"	32
277	"te"	116
278	"**ea**s"	264

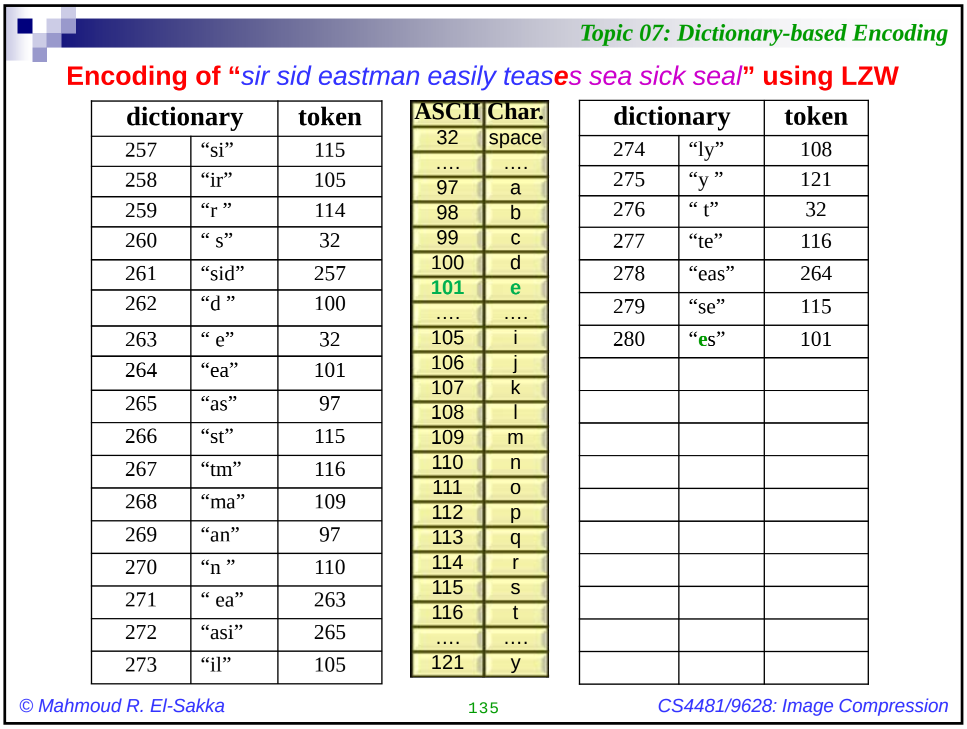


Encoding of "*sir sid eastman easily teases sea sick seal*" using LZW

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	"ea"	101
265	"as"	97
266	"st"	115
267	"tm"	116
268	"ma"	109
269	"an"	97
270	"n "	110
271	" ea"	263
272	"asi"	265
273	"il"	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	**s**
116	t
….	….
121	y

dictionary		token
274	"ly"	108
275	"y "	121
276	" t"	32
277	"te"	116
278	"eas"	264
279	"**s**e"	115



Topic 07: Dictionary-based Encoding

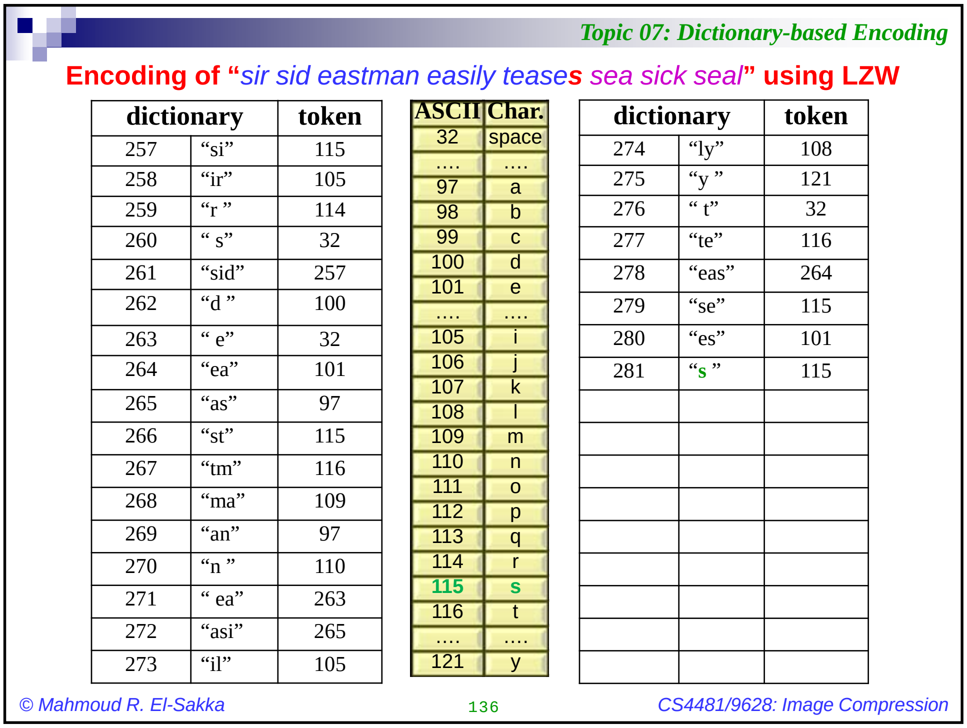
Encoding of "sir sid eastman easily teas**e**s sea sick seal" using LZW

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	"ea"	101
265	"as"	97
266	"st"	115
267	"tm"	116
268	"ma"	109
269	"an"	97
270	"n "	110
271	" ea"	263
272	"asi"	265
273	"il"	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	"ly"	108
275	"y "	121
276	" t"	32
277	"te"	116
278	"eas"	264
279	"se"	115
280	"es"	101

© Mahmoud R. El-Sakka 135 CS4481/9628: Image Compression



Topic 07: Dictionary-based Encoding

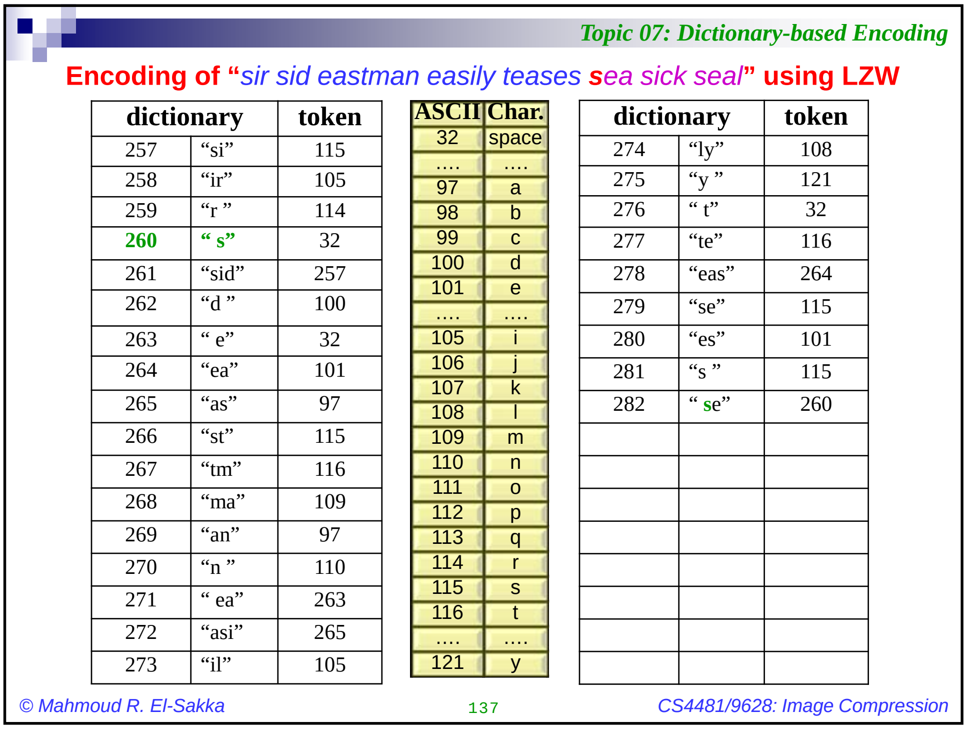
Encoding of "sir sid eastman easily tease**s** sea sick seal" using LZW

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	"ea"	101
265	"as"	97
266	"st"	115
267	"tm"	116
268	"ma"	109
269	"an"	97
270	"n "	110
271	" ea"	263
272	"asi"	265
273	"il"	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	"ly"	108
275	"y "	121
276	" t"	32
277	"te"	116
278	"eas"	264
279	"se"	115
280	"es"	101
281	"s "	115

© Mahmoud R. El-Sakka 136 CS4481/9628: Image Compression

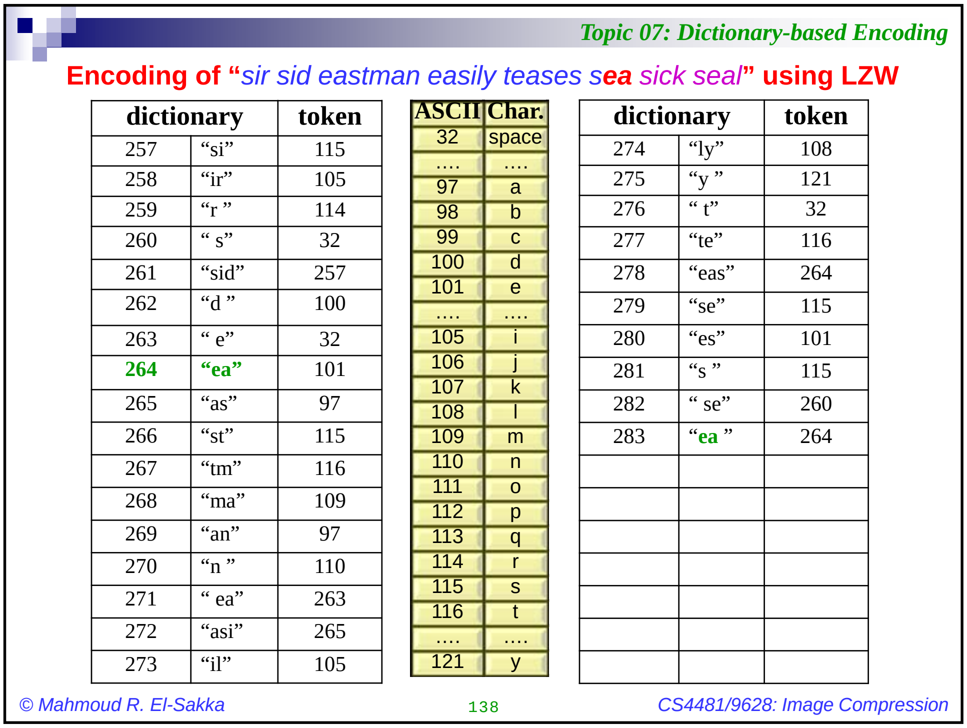


Encoding of “sir sid eastman easily teases sea sick seal” using LZW

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	**“ s”**	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“asi”	265
273	“il”	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	“ly”	108
275	“y ”	121
276	“ t”	32
277	“te”	116
278	“eas”	264
279	“se”	115
280	“es”	101
281	“s ”	115
282	“ **se**”	260

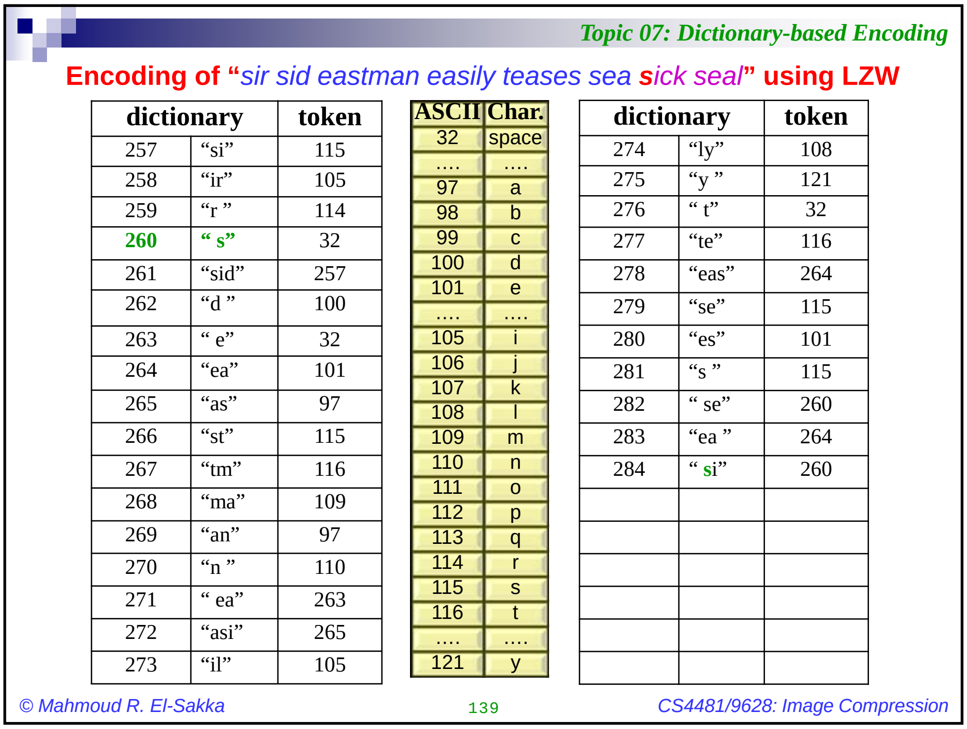


Encoding of “sir sid eastman easily teases sea sick seal” using LZW

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	**“ea”**	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“asi”	265
273	“il”	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	“ly”	108
275	“y ”	121
276	“ t”	32
277	“te”	116
278	“eas”	264
279	“se”	115
280	“es”	101
281	“s ”	115
282	“ se”	260
283	“**ea** ”	264

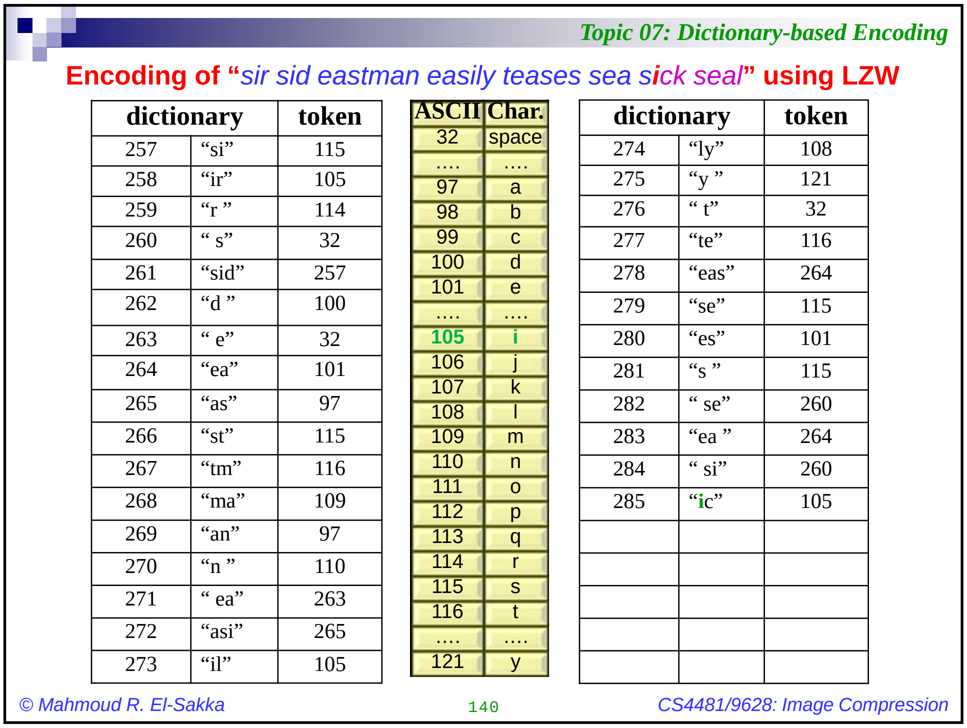


Encoding of "*sir sid eastman easily teases sea sick seal*" using LZW

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	**" s"**	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	"ea"	101
265	"as"	97
266	"st"	115
267	"tm"	116
268	"ma"	109
269	"an"	97
270	"n "	110
271	" ea"	263
272	"asi"	265
273	"il"	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	"ly"	108
275	"y "	121
276	" t"	32
277	"te"	116
278	"eas"	264
279	"se"	115
280	"es"	101
281	"s "	115
282	" se"	260
283	"ea "	264
284	" **si**"	260

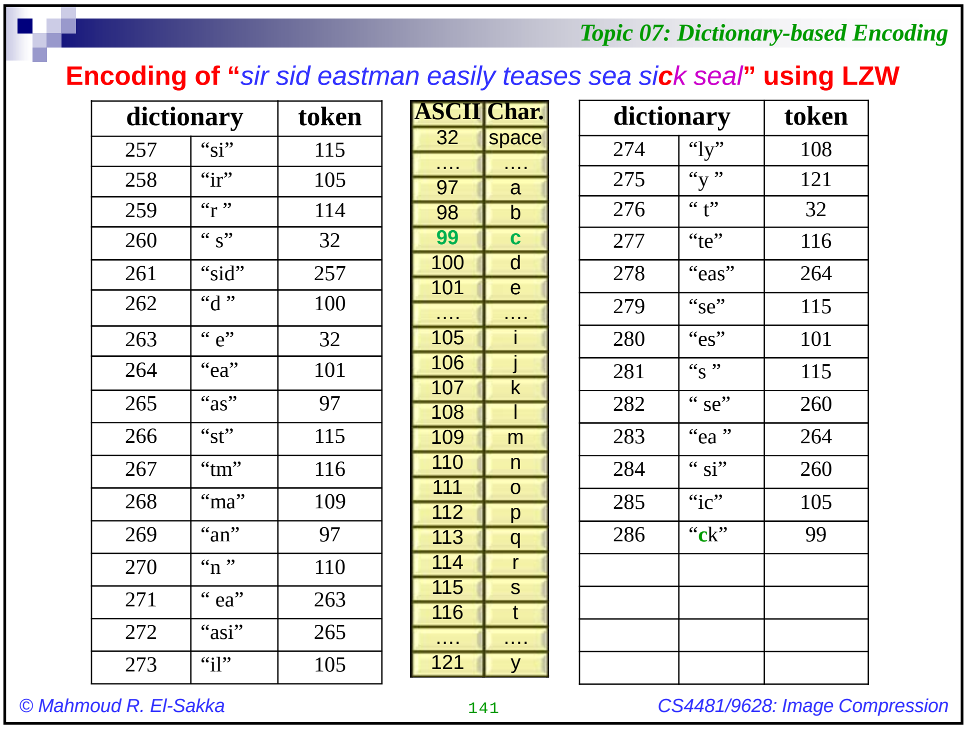


Encoding of "*sir sid eastman easily teases sea sick seal*" using LZW

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	"ea"	101
265	"as"	97
266	"st"	115
267	"tm"	116
268	"ma"	109
269	"an"	97
270	"n "	110
271	" ea"	263
272	"asi"	265
273	"il"	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	**i**
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	"ly"	108
275	"y "	121
276	" t"	32
277	"te"	116
278	"eas"	264
279	"se"	115
280	"es"	101
281	"s "	115
282	" se"	260
283	"ea "	264
284	" si"	260
285	"**i**c"	105

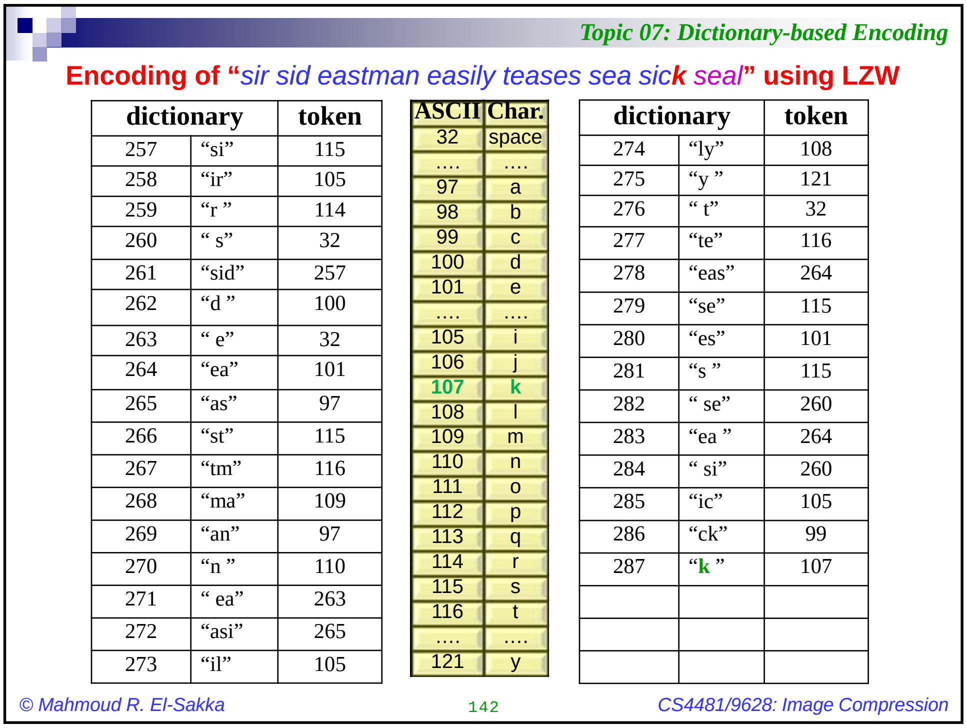


Encoding of “sir sid eastman easily teases sea sick seal” using LZW

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“asi”	265
273	“il”	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	“ly”	108
275	“y ”	121
276	“ t”	32
277	“te”	116
278	“eas”	264
279	“se”	115
280	“es”	101
281	“s ”	115
282	“ se”	260
283	“ea ”	264
284	“ si”	260
285	“ic”	105
286	“ck”	99

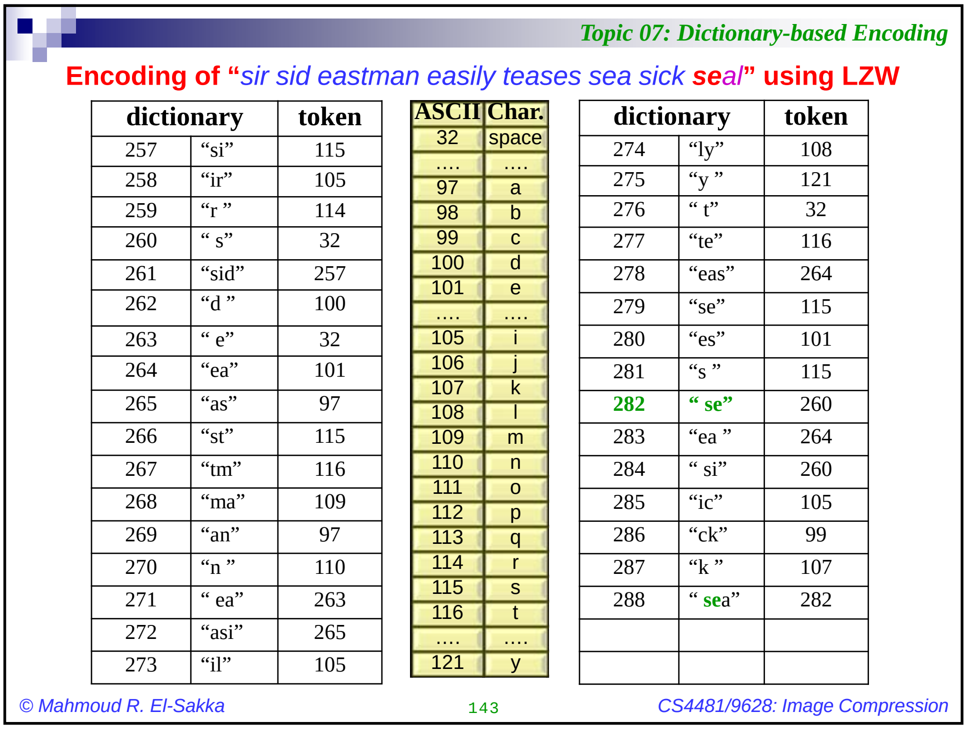


Encoding of “sir sid eastman easily teases sea sick seal” using LZW

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“asi”	265
273	“il”	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	“ly”	108
275	“y ”	121
276	“ t”	32
277	“te”	116
278	“eas”	264
279	“se”	115
280	“es”	101
281	“s ”	115
282	“ se”	260
283	“ea ”	264
284	“ si”	260
285	“ic”	105
286	“ck”	99
287	“k ”	107

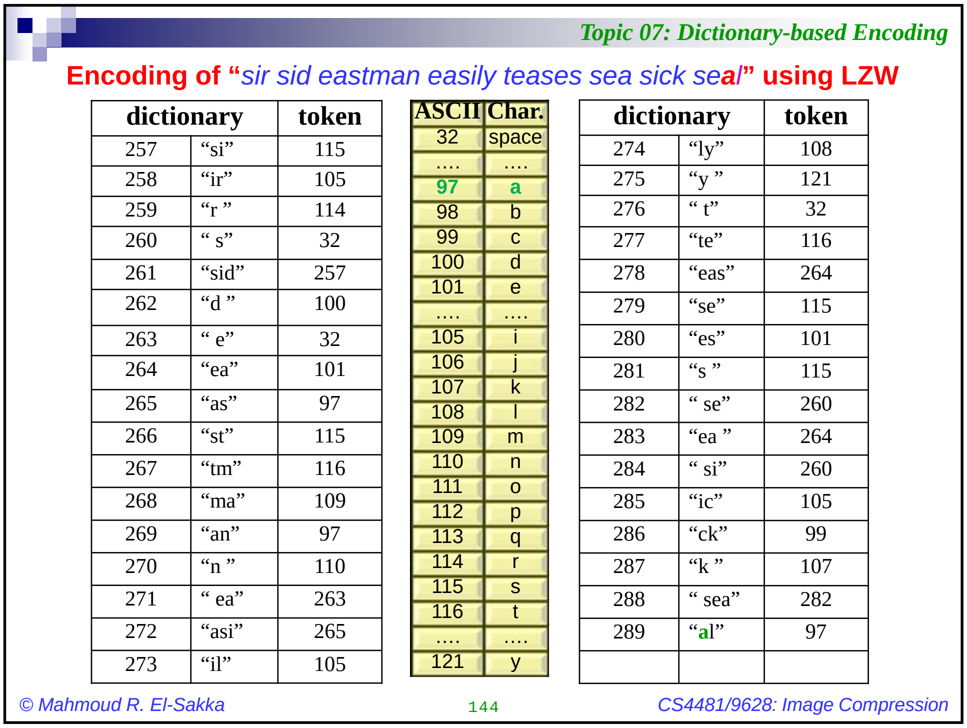


Encoding of "sir sid eastman easily teases sea sick seal" using LZW

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	"ea"	101
265	"as"	97
266	"st"	115
267	"tm"	116
268	"ma"	109
269	"an"	97
270	"n "	110
271	" ea"	263
272	"asi"	265
273	"il"	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	"ly"	108
275	"y "	121
276	" t"	32
277	"te"	116
278	"eas"	264
279	"se"	115
280	"es"	101
281	"s "	115
282	**" se"**	260
283	"ea "	264
284	" si"	260
285	"ic"	105
286	"ck"	99
287	"k "	107
288	" **sea**"	282



Encoding of "sir sid eastman easily teases sea sick seal" using LZW

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	"ea"	101
265	"as"	97
266	"st"	115
267	"tm"	116
268	"ma"	109
269	"an"	97
270	"n "	110
271	" ea"	263
272	"asi"	265
273	"il"	105

ASCII	Char.
32	space
….	….
97	**a**
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	"ly"	108
275	"y "	121
276	" t"	32
277	"te"	116
278	"eas"	264
279	"se"	115
280	"es"	101
281	"s "	115
282	" se"	260
283	"ea "	264
284	" si"	260
285	"ic"	105
286	"ck"	99
287	"k "	107
288	" sea"	282
289	"**al**"	97

Encoding of "*sir sid eastman easily teases sea sick seal*" using LZW

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“asi”	265
273	“il”	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	**l**
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	“ly”	108
275	“y ”	121
276	“ t”	32
277	“te”	116
278	“eas”	264
279	“se”	115
280	“es”	101
281	“s ”	115
282	“ se”	260
283	“ea ”	264
284	“ si”	260
285	“ic”	105
286	“ck”	99
287	“k ”	107
288	“ sea”	282
289	“al”	97
		108

Decoding using LZW

“”

dictionary		token
257		115
258		105
259		114
260		32
261		257
262		100
263		32
264		101
265		97
266		115
267		116
268		109
269		97
270		110
271		263
272		265
273		105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274		108
275		121
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108

Decoding using LZW

"s"

dictionary		token
257		115
258		105
259		114
260		32
261		257
262		100
263		32
264		101
265		97
266		115
267		116
268		109
269		97
270		110
271		263
272		265
273		105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274		108
275		121
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108

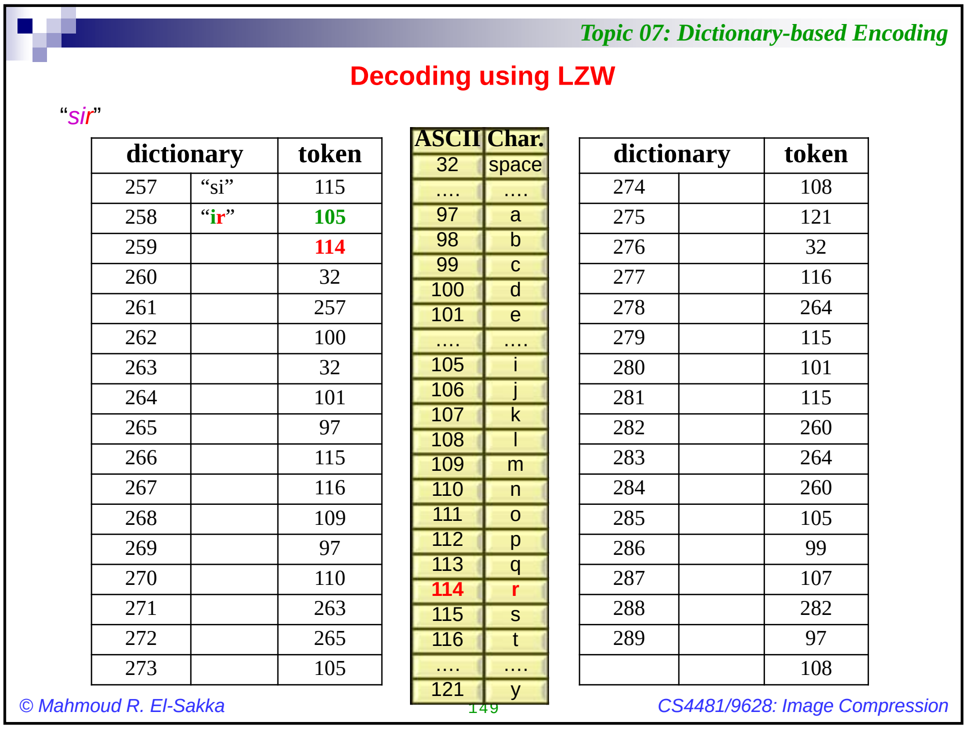
Decoding using LZW

"si"

dictionary		token
257	"si"	115
258		105
259		114
260		32
261		257
262		100
263		32
264		101
265		97
266		115
267		116
268		109
269		97
270		110
271		263
272		265
273		105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274		108
275		121
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108



Topic 07: Dictionary-based Encoding

Decoding using LZW

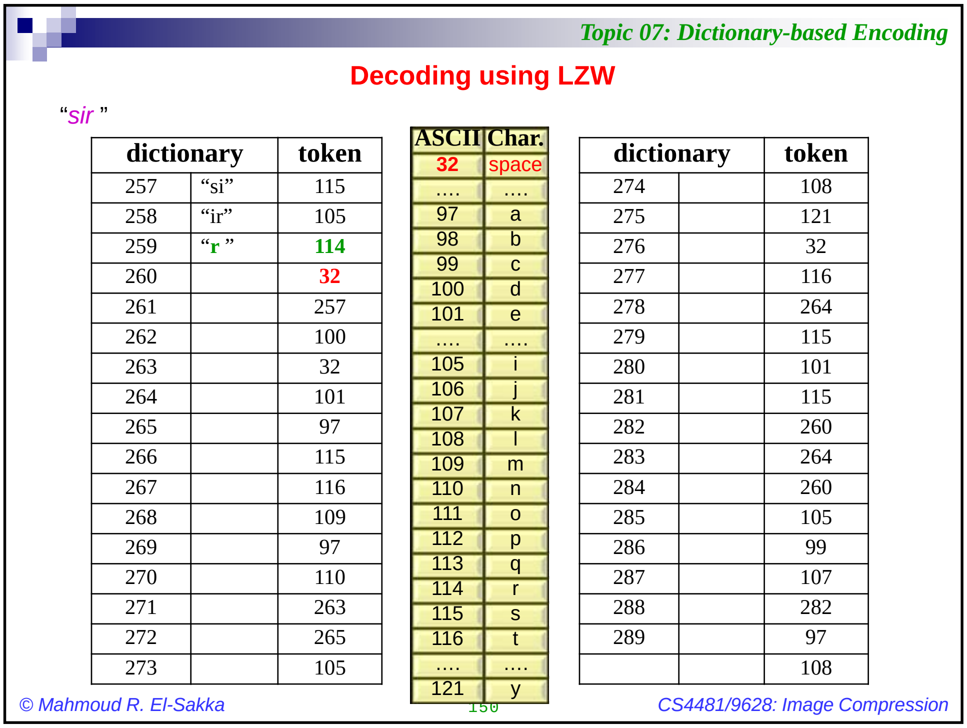
"sir"

dictionary		token
257	"si"	115
258	"ir"	105
259		114
260		32
261		257
262		100
263		32
264		101
265		97
266		115
267		116
268		109
269		97
270		110
271		263
272		265
273		105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274		108
275		121
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108

© Mahmoud R. El-Sakka 149 CS4481/9628: Image Compression



Topic 07: Dictionary-based Encoding

Decoding using LZW

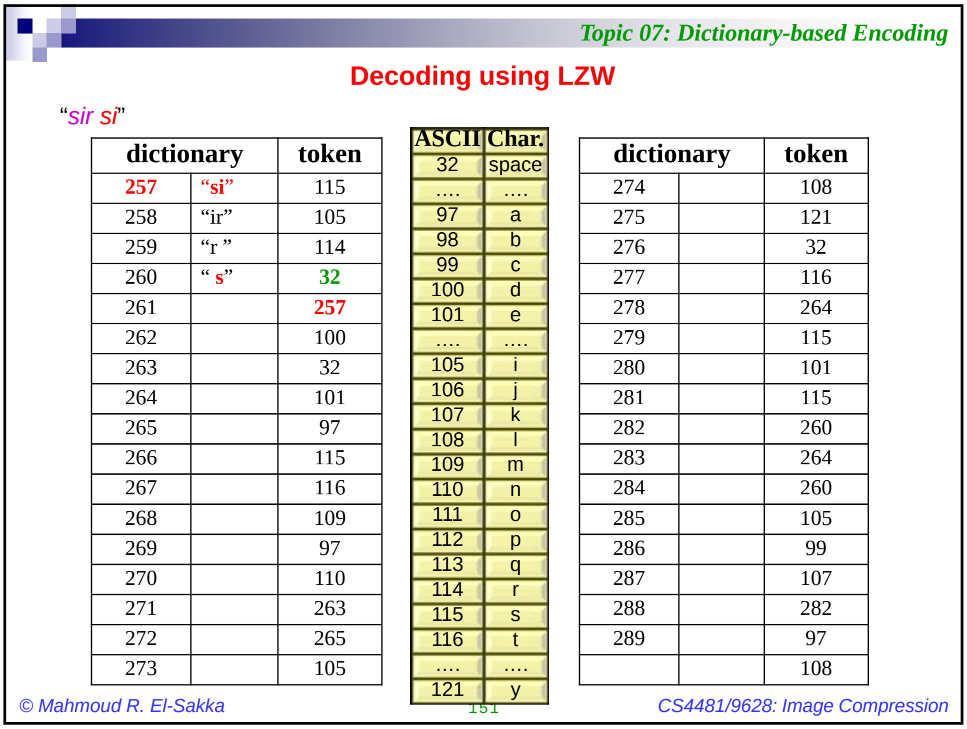
"sir "

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260		32
261		257
262		100
263		32
264		101
265		97
266		115
267		116
268		109
269		97
270		110
271		263
272		265
273		105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274		108
275		121
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108

© Mahmoud R. El-Sakka 150 CS4481/9628: Image Compression



Topic 07: Dictionary-based Encoding

Decoding using LZW

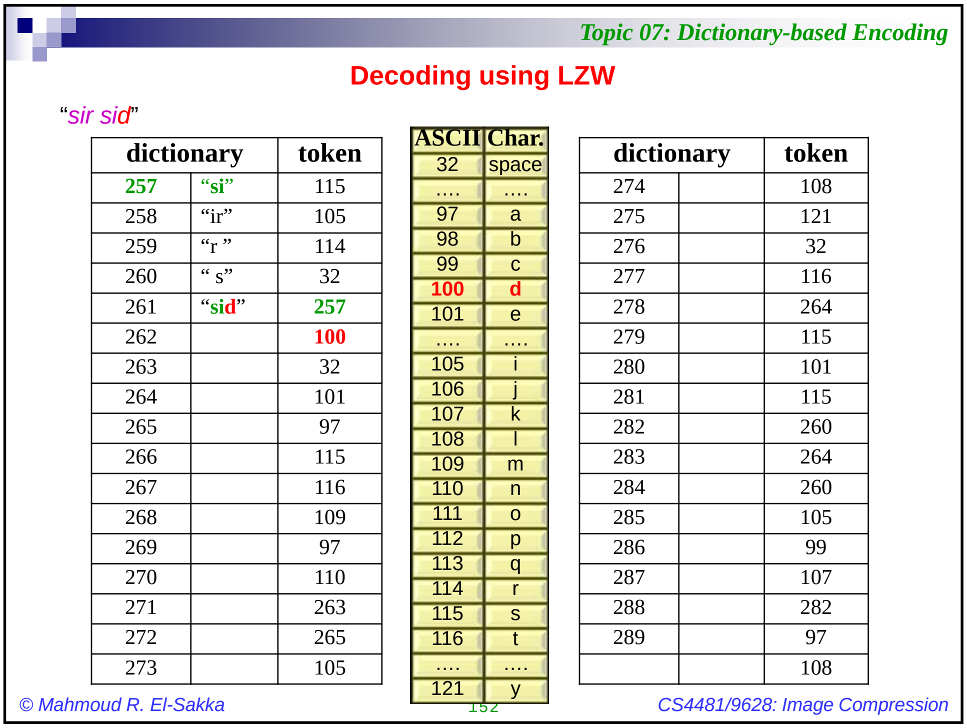
"*sir si*"

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261		257
262		100
263		32
264		101
265		97
266		115
267		116
268		109
269		97
270		110
271		263
272		265
273		105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274		108
275		121
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108

© Mahmoud R. El-Sakka 151 CS4481/9628: Image Compression



Topic 07: Dictionary-based Encoding

Decoding using LZW

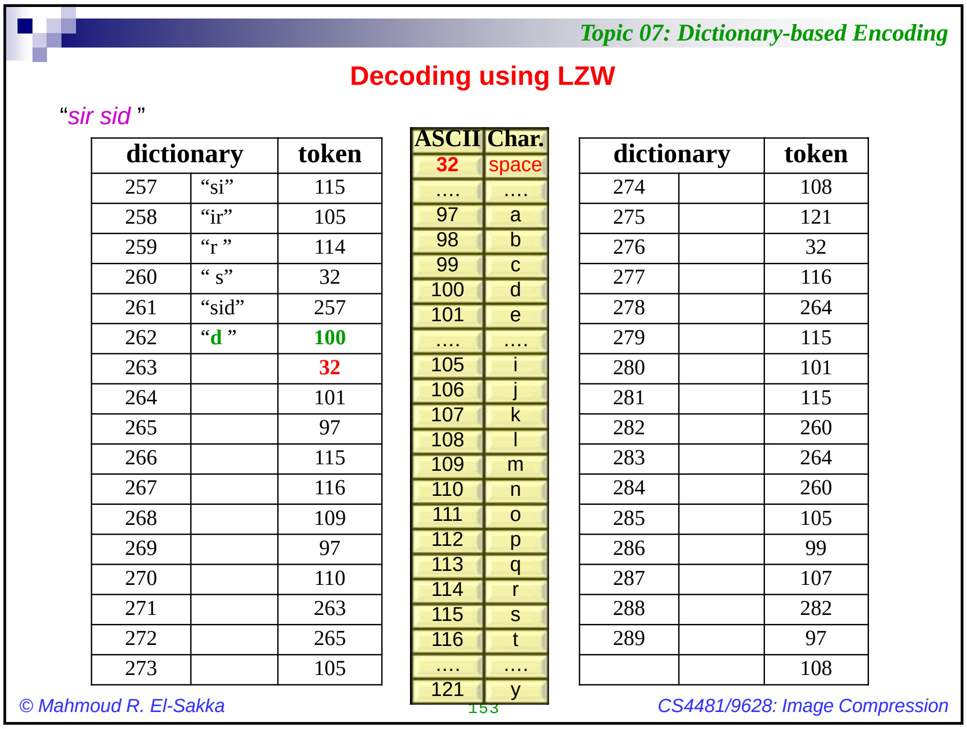
"*sir sid*"

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262		100
263		32
264		101
265		97
266		115
267		116
268		109
269		97
270		110
271		263
272		265
273		105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274		108
275		121
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108

© Mahmoud R. El-Sakka 152 CS4481/9628: Image Compression



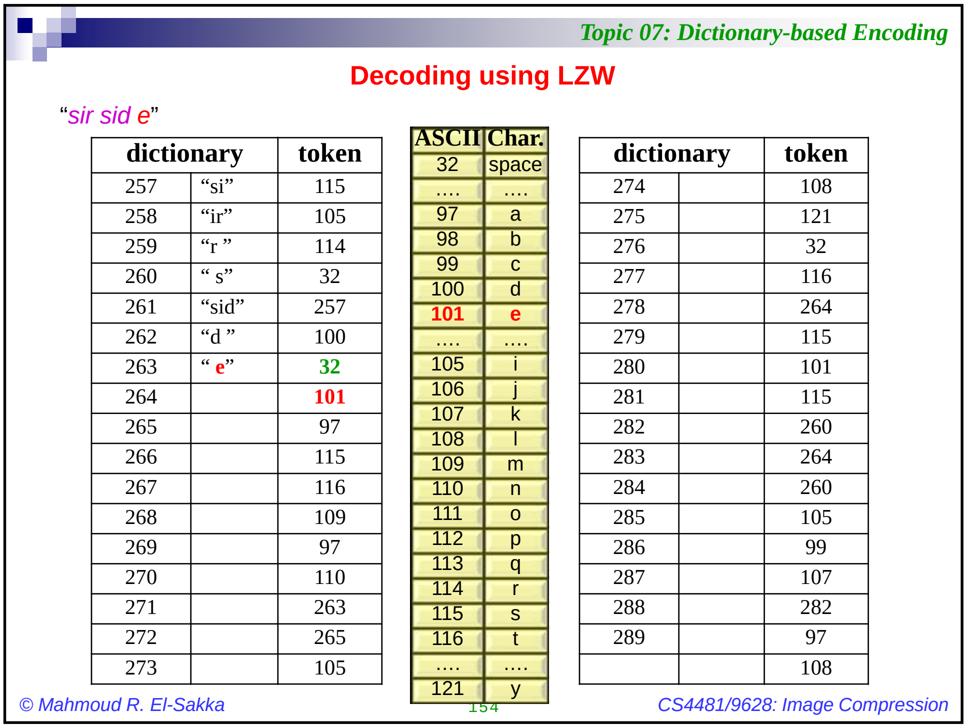
Decoding using LZW

“sir sid ”

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263		32
264		101
265		97
266		115
267		116
268		109
269		97
270		110
271		263
272		265
273		105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274		108
275		121
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108



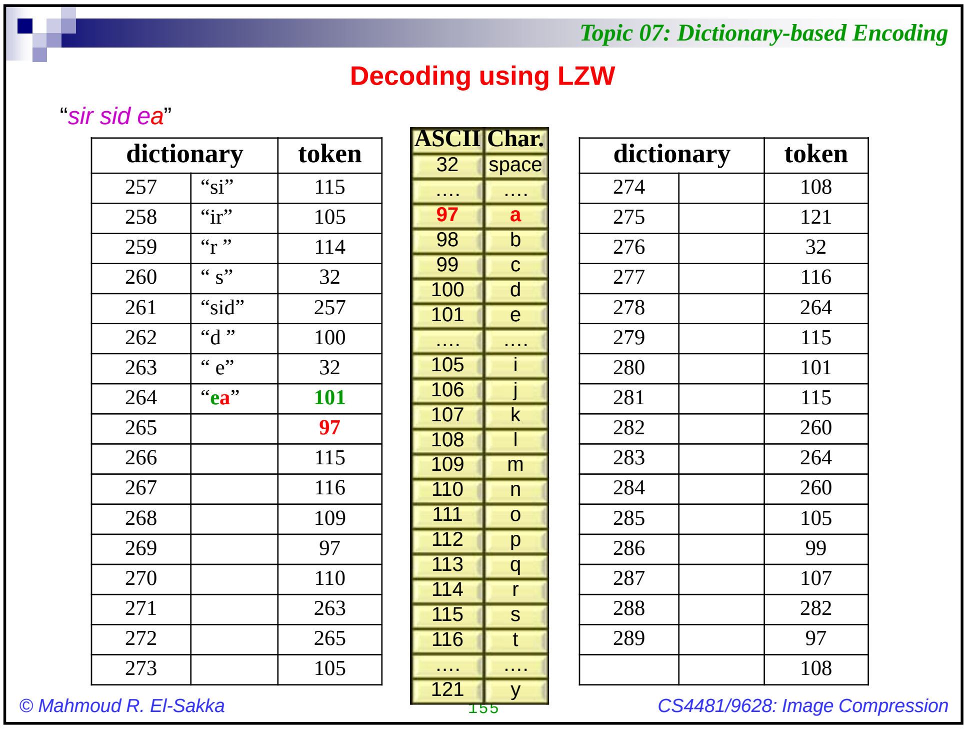
Decoding using LZW

“sir sid e”

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264		101
265		97
266		115
267		116
268		109
269		97
270		110
271		263
272		265
273		105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274		108
275		121
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108



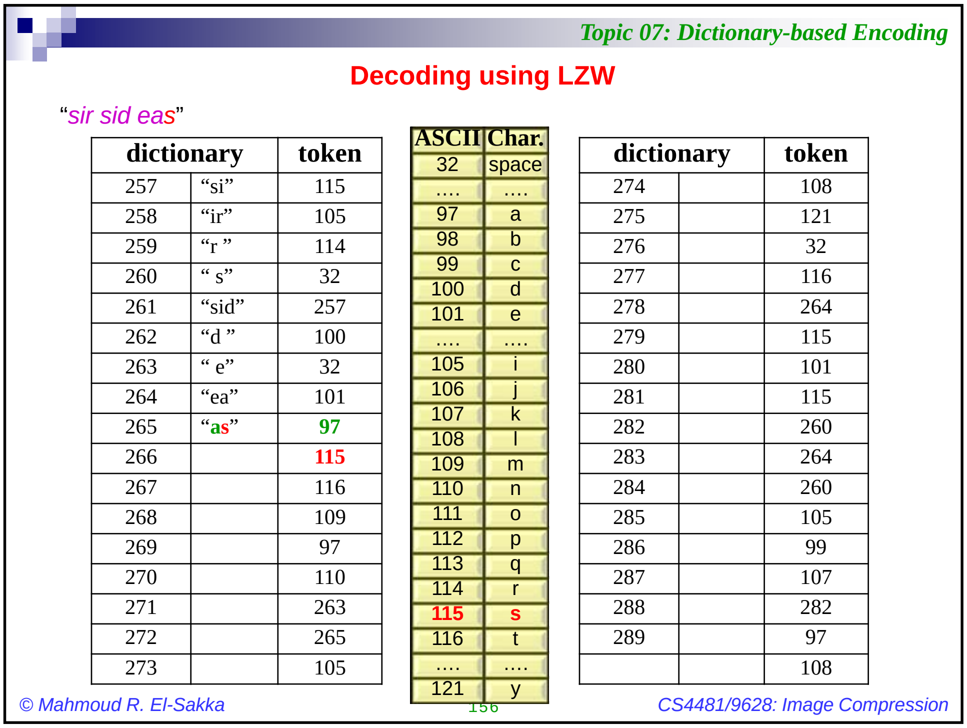
Decoding using LZW

"*sir sid ea*"

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	"**ea**"	**101**
265		**97**
266		115
267		116
268		109
269		97
270		110
271		263
272		265
273		105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274		108
275		121
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108



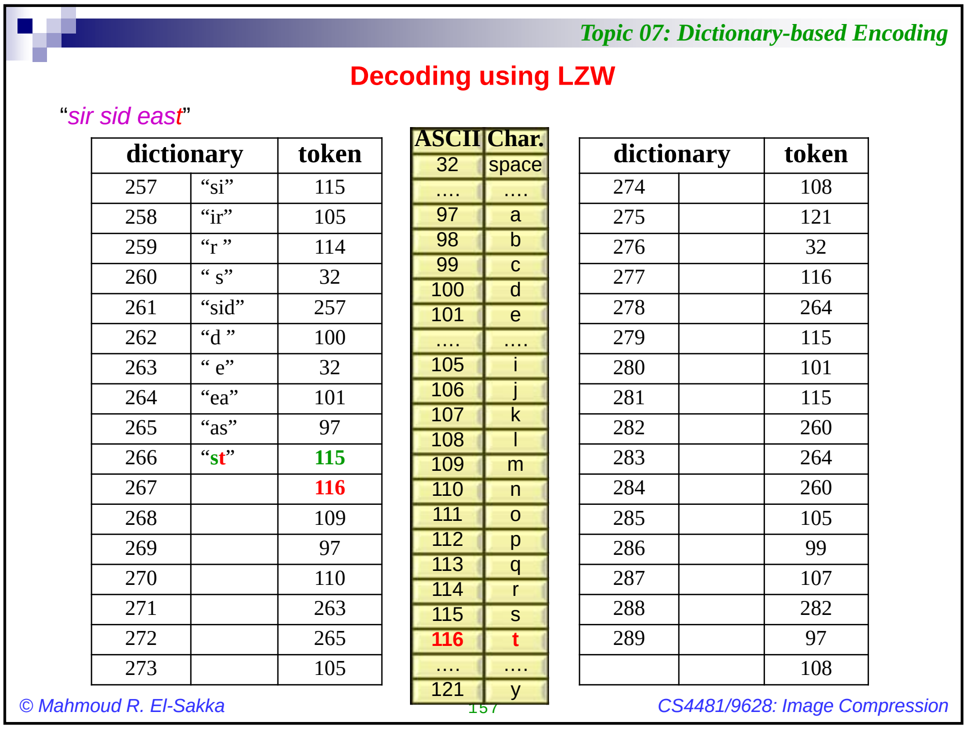
Decoding using LZW

"*sir sid eas*"

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	"ea"	101
265	"**as**"	**97**
266		**115**
267		116
268		109
269		97
270		110
271		263
272		265
273		105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274		108
275		121
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108



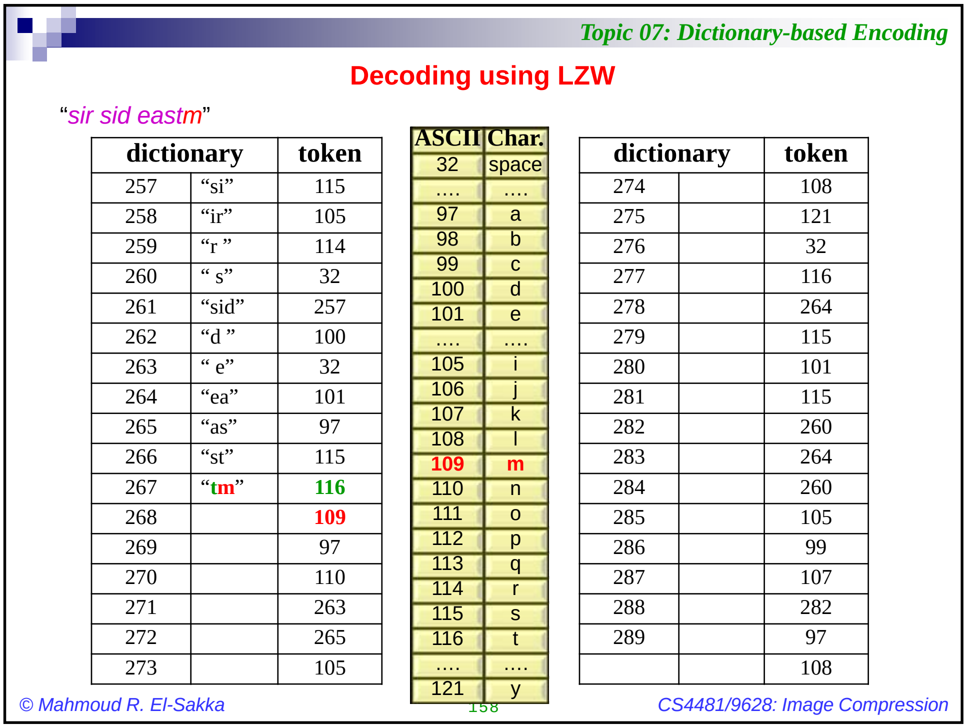
Decoding using LZW

"*sir sid east*"

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	"ea"	101
265	"as"	97
266	"**st**"	**115**
267		**116**
268		109
269		97
270		110
271		263
272		265
273		105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	**t**
….	….
121	y

dictionary		token
274		108
275		121
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108



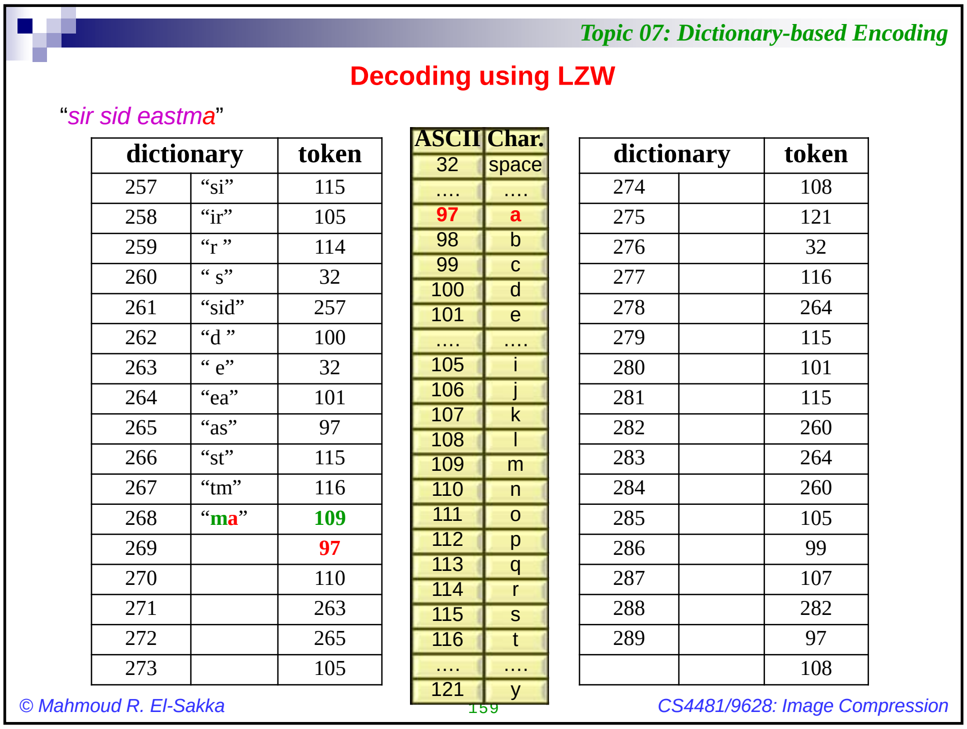
Decoding using LZW

"*sir sid eastm*"

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	"ea"	101
265	"as"	97
266	"st"	115
267	"**tm**"	**116**
268		**109**
269		97
270		110
271		263
272		265
273		105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	**m**
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274		108
275		121
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108



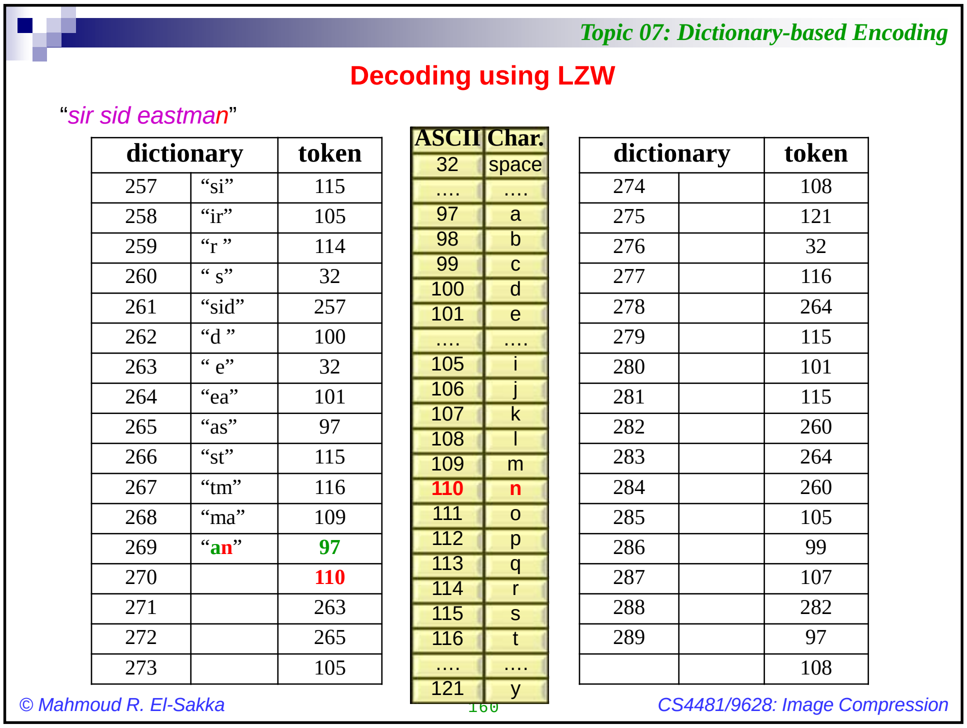
Decoding using LZW

"sir sid eastma"

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	"ea"	101
265	"as"	97
266	"st"	115
267	"tm"	116
268	"**ma**"	**109**
269		**97**
270		110
271		263
272		265
273		105

ASCII	Char.
32	space
….	….
97	**a**
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274		108
275		121
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108



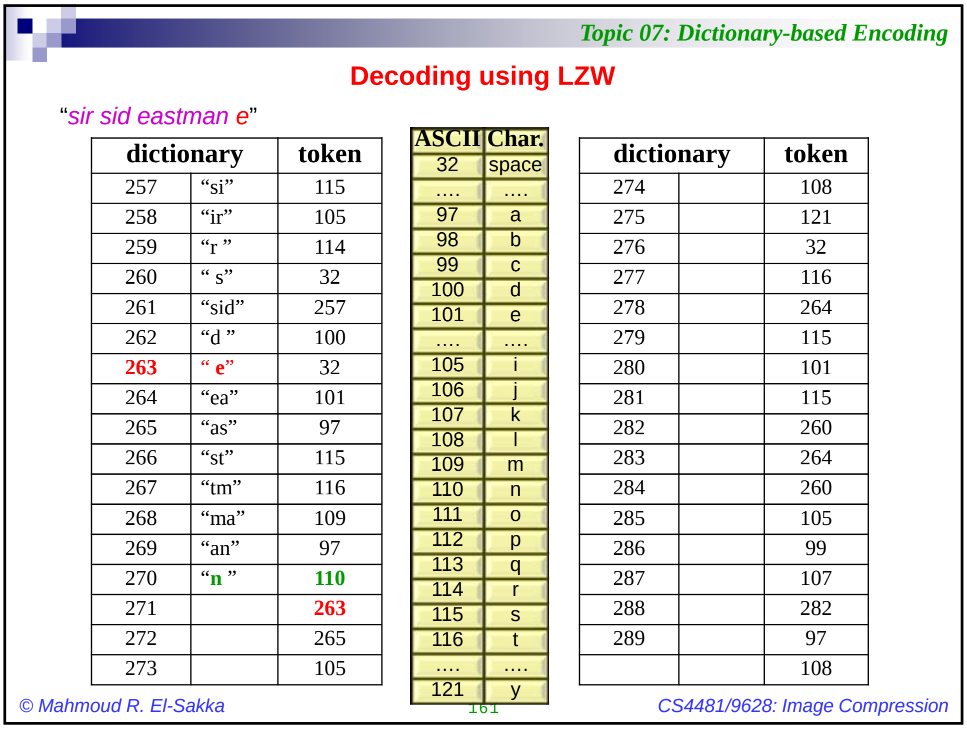
Decoding using LZW

"sir sid eastman"

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	"ea"	101
265	"as"	97
266	"st"	115
267	"tm"	116
268	"ma"	109
269	"**an**"	**97**
270		**110**
271		263
272		265
273		105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	**n**
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274		108
275		121
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108



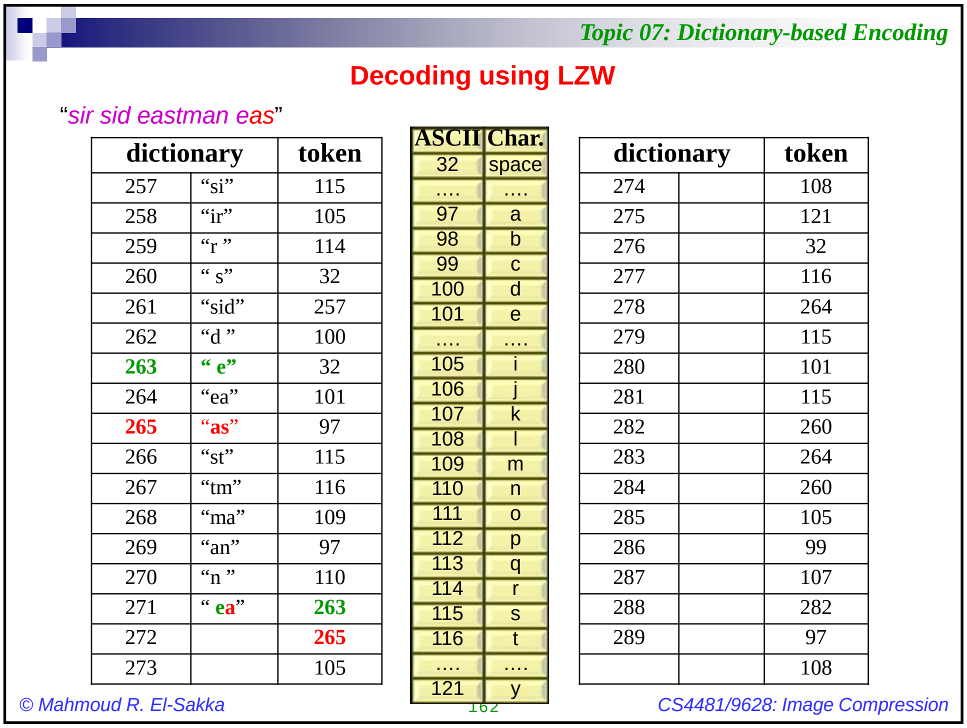
Decoding using LZW

"sir sid eastman e"

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	**" e"**	32
264	"ea"	101
265	"as"	97
266	"st"	115
267	"tm"	116
268	"ma"	109
269	"an"	97
270	**"n "**	**110**
271		**263**
272		265
273		105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274		108
275		121
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108



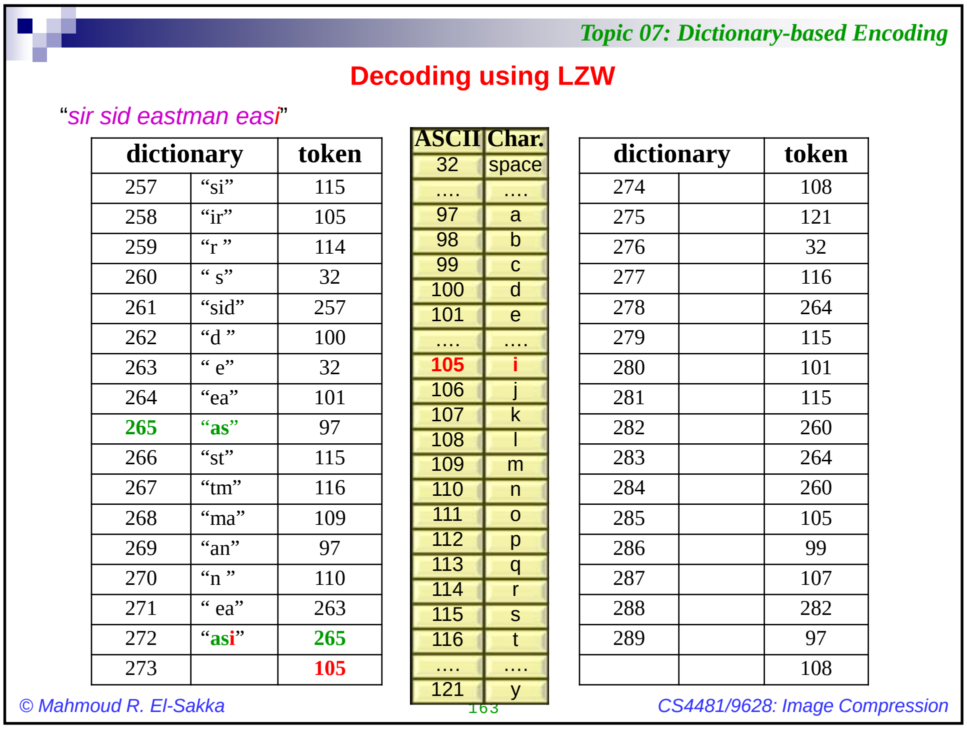
Decoding using LZW

"sir sid eastman eas"

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	**" e"**	32
264	"ea"	101
265	**"as"**	97
266	"st"	115
267	"tm"	116
268	"ma"	109
269	"an"	97
270	"n "	110
271	**" ea"**	**263**
272		**265**
273		105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274		108
275		121
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108



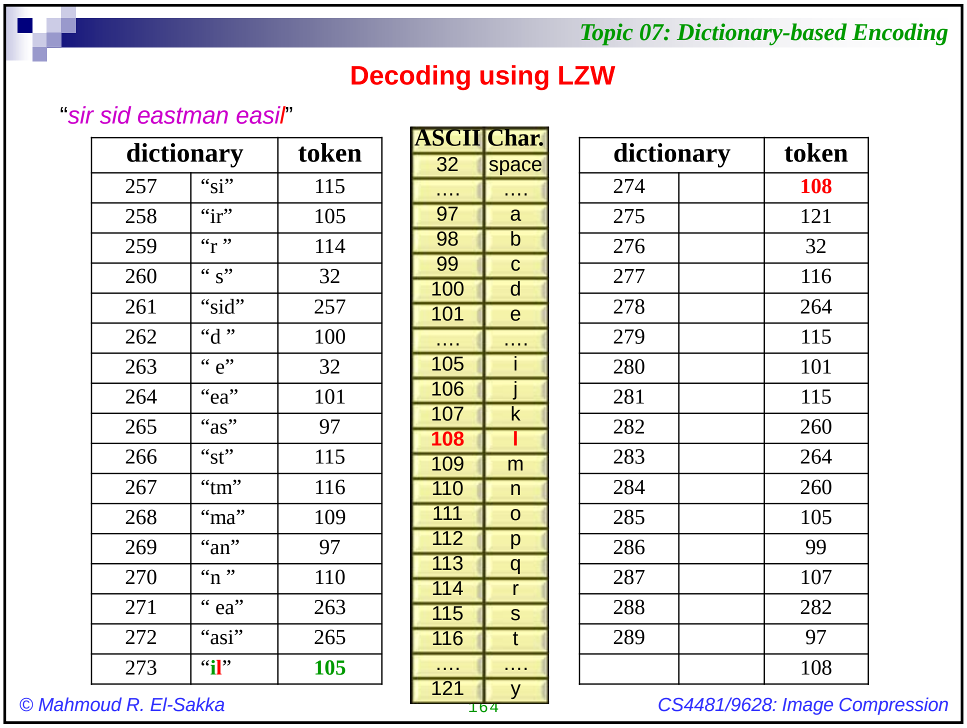
Decoding using LZW

“*sir sid eastman easi*”

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	**“as”**	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“**asi**”	**265**
273		**105**

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	**i**
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274		108
275		121
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108



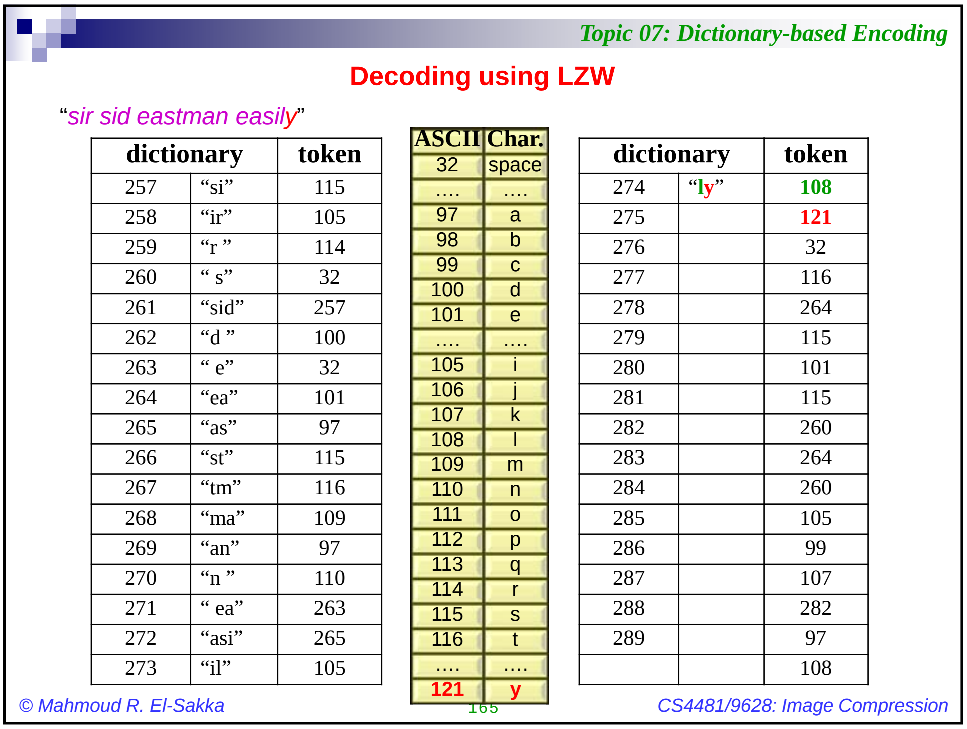
Decoding using LZW

“*sir sid eastman easil*”

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“asi”	265
273	“**il**”	**105**

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	**l**
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274		**108**
275		121
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108



Topic 07: Dictionary-based Encoding

Decoding using LZW

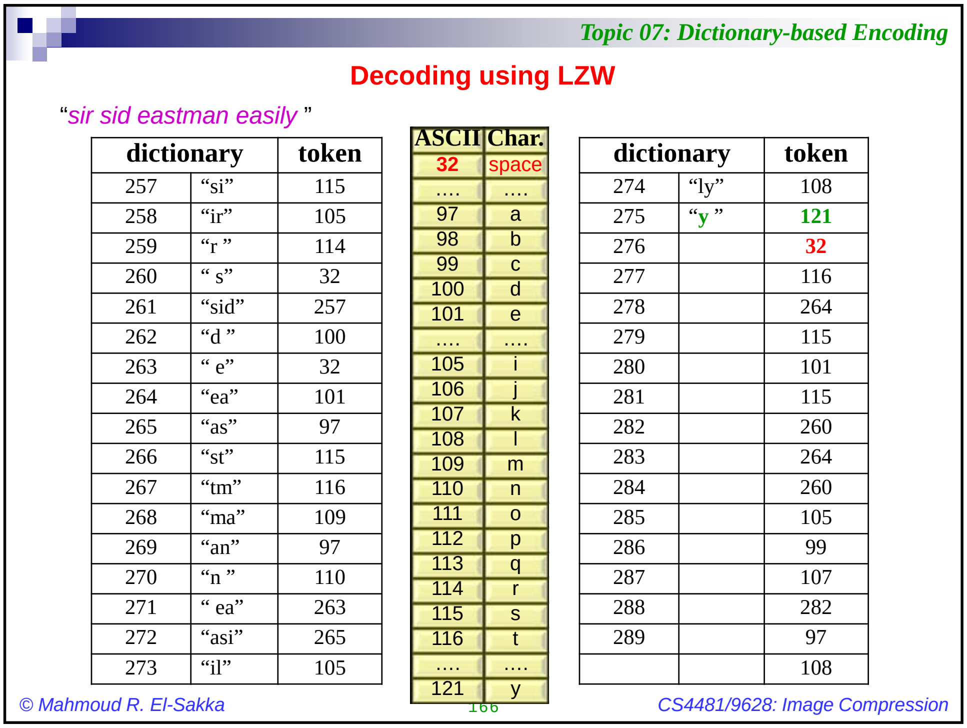
"*sir sid eastman easily*"

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	"ea"	101
265	"as"	97
266	"st"	115
267	"tm"	116
268	"ma"	109
269	"an"	97
270	"n "	110
271	" ea"	263
272	"asi"	265
273	"il"	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	**y**

dictionary		token
274	"ly"	**108**
275		**121**
276		32
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108

© Mahmoud R. El-Sakka 165 CS4481/9628: Image Compression



Topic 07: Dictionary-based Encoding

Decoding using LZW

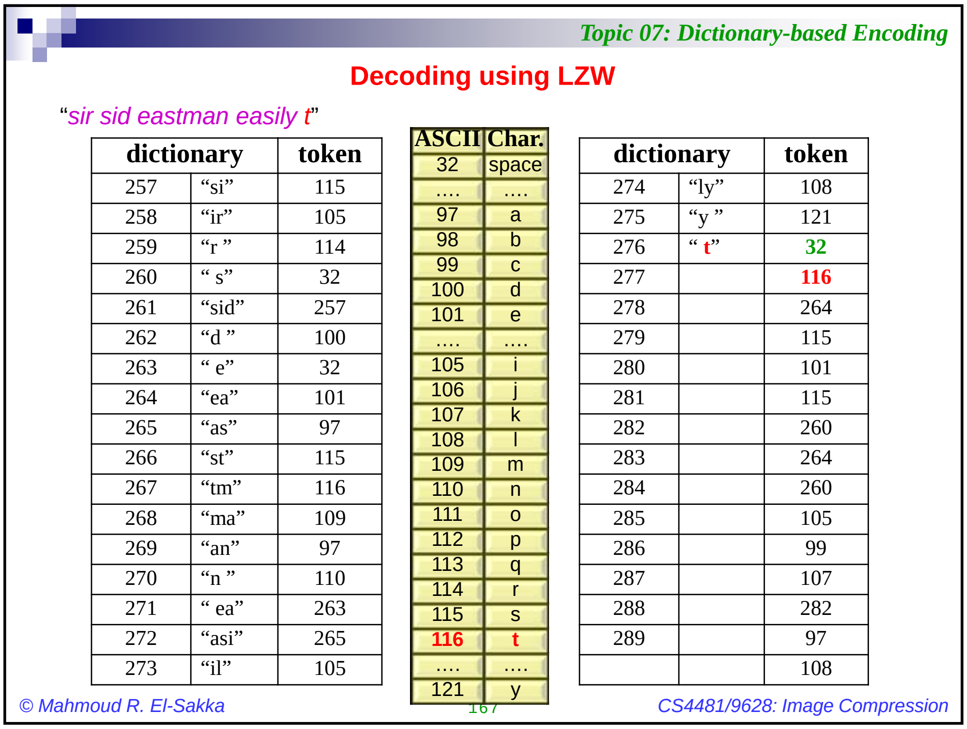
"*sir sid eastman easily* "

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	"ea"	101
265	"as"	97
266	"st"	115
267	"tm"	116
268	"ma"	109
269	"an"	97
270	"n "	110
271	" ea"	263
272	"asi"	265
273	"il"	105

ASCII	Char.
32	**space**
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	"ly"	108
275	"y "	**121**
276		**32**
277		116
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108

© Mahmoud R. El-Sakka 166 CS4481/9628: Image Compression



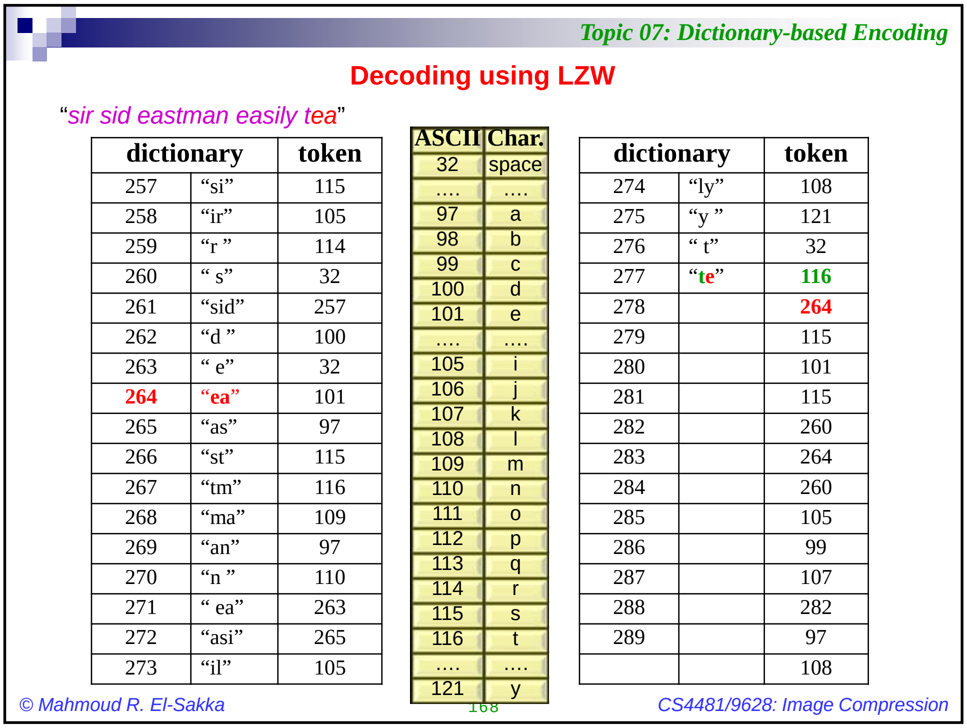
Decoding using LZW

“sir sid eastman easily t”

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“asi”	265
273	“il”	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	**t**
….	….
121	y

dictionary		token
274	“ly”	108
275	“y ”	121
276	“ **t**”	**32**
277		**116**
278		264
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108



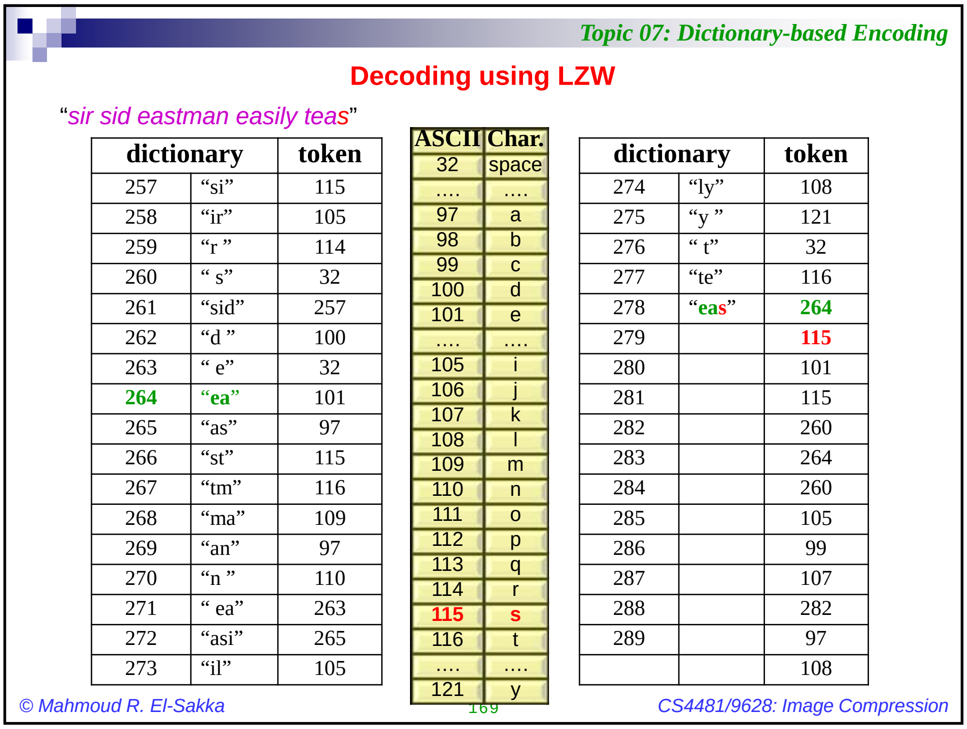
Decoding using LZW

“sir sid eastman easily tea”

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	**“ea”**	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“asi”	265
273	“il”	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	“ly”	108
275	“y ”	121
276	“ t”	32
277	“**te**”	**116**
278		**264**
279		115
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108



Topic 07: Dictionary-based Encoding

Decoding using LZW

"*sir sid eastman easily teas*"

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	**"ea"**	101
265	"as"	97
266	"st"	115
267	"tm"	116
268	"ma"	109
269	"an"	97
270	"n "	110
271	" ea"	263
272	"asi"	265
273	"il"	105

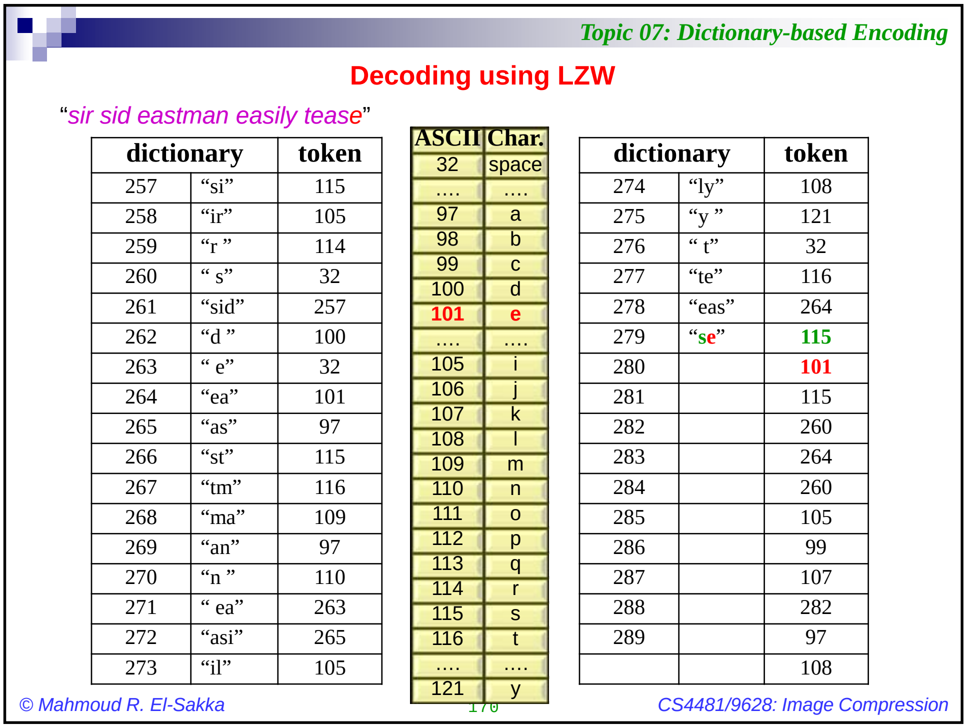
ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	**s**
116	t
….	….
121	y

dictionary		token
274	"ly"	108
275	"y "	121
276	" t"	32
277	"te"	116
278	**"eas"**	**264**
279		**115**
280		101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108

© Mahmoud R. El-Sakka

169

CS4481/9628: Image Compression



Topic 07: Dictionary-based Encoding

Decoding using LZW

"*sir sid eastman easily tease*"

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	" s"	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	"ea"	101
265	"as"	97
266	"st"	115
267	"tm"	116
268	"ma"	109
269	"an"	97
270	"n "	110
271	" ea"	263
272	"asi"	265
273	"il"	105

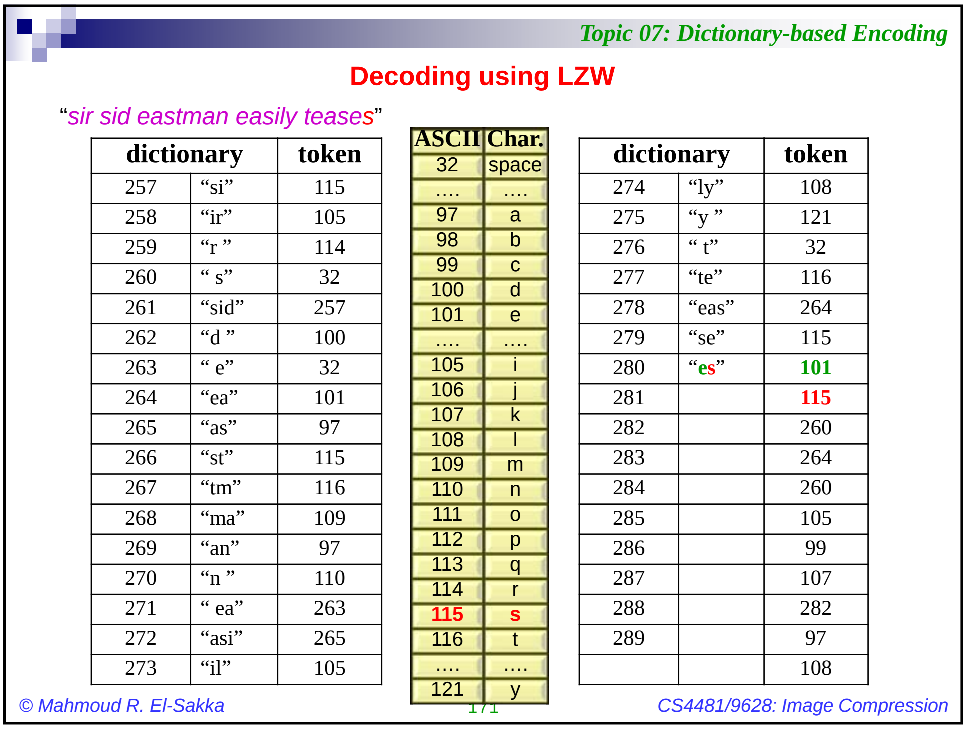
ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	**e**
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	"ly"	108
275	"y "	121
276	" t"	32
277	"te"	116
278	"eas"	264
279	**"se"**	**115**
280		**101**
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108

© Mahmoud R. El-Sakka

170

CS4481/9628: Image Compression



Topic 07: Dictionary-based Encoding

Decoding using LZW

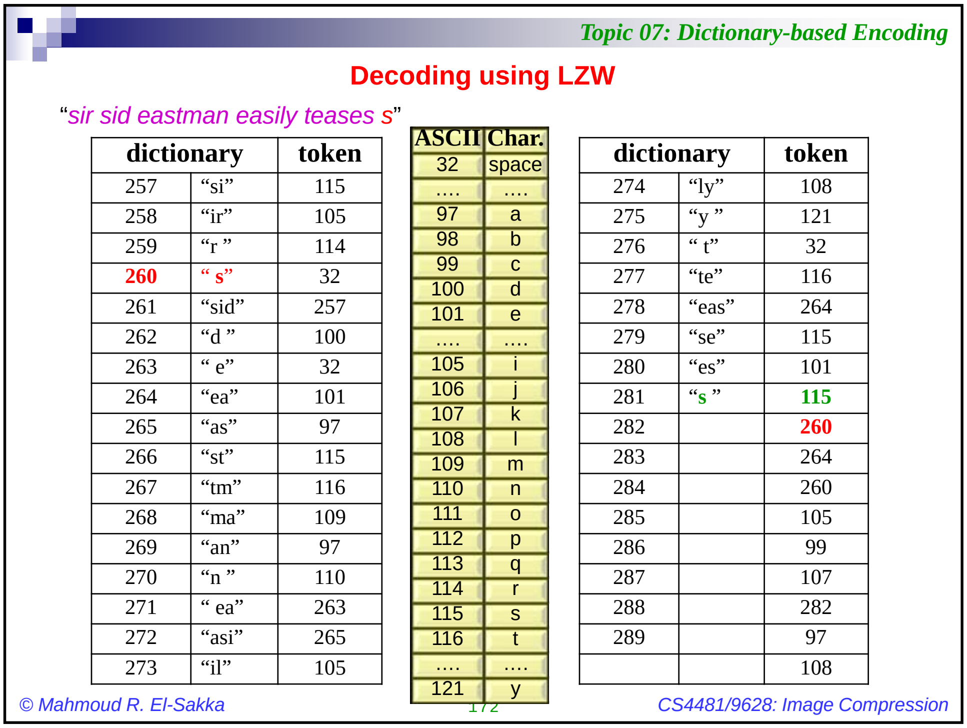
“*sir sid eastman easily teases*”

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“asi”	265
273	“il”	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	“ly”	108
275	“y ”	121
276	“ t”	32
277	“te”	116
278	“eas”	264
279	“se”	115
280	“es”	101
281		115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108

© Mahmoud R. El-Sakka 171 CS4481/9628: Image Compression



Topic 07: Dictionary-based Encoding

Decoding using LZW

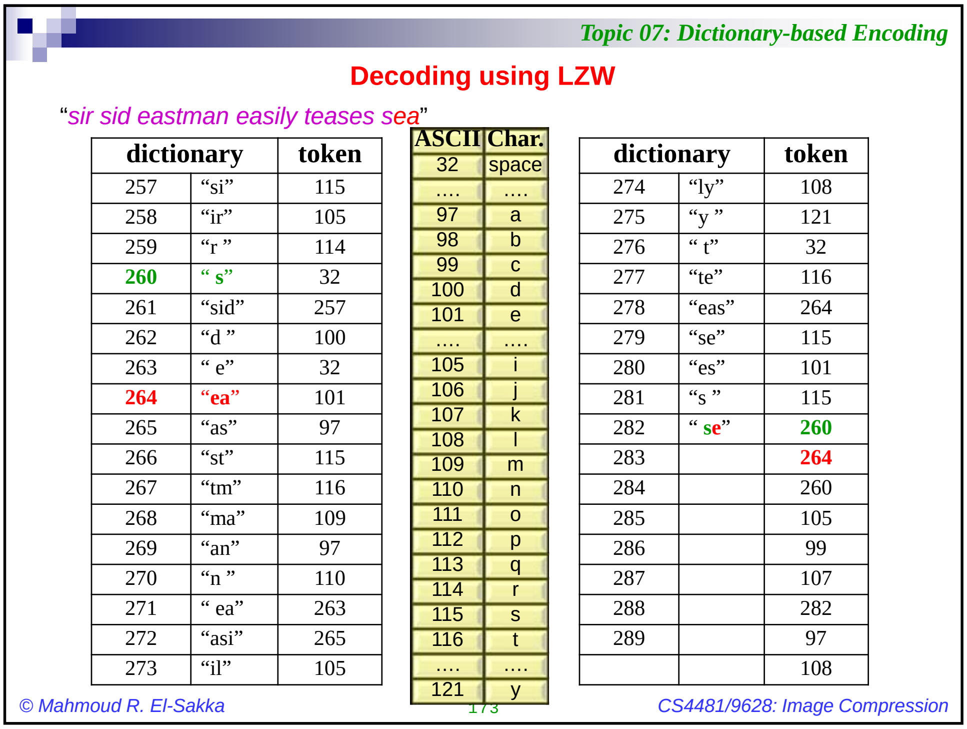
“*sir sid eastman easily teases s*”

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“asi”	265
273	“il”	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	“ly”	108
275	“y ”	121
276	“ t”	32
277	“te”	116
278	“eas”	264
279	“se”	115
280	“es”	101
281	“s ”	115
282		260
283		264
284		260
285		105
286		99
287		107
288		282
289		97
		108

© Mahmoud R. El-Sakka 172 CS4481/9628: Image Compression



Topic 07: Dictionary-based Encoding

Decoding using LZW

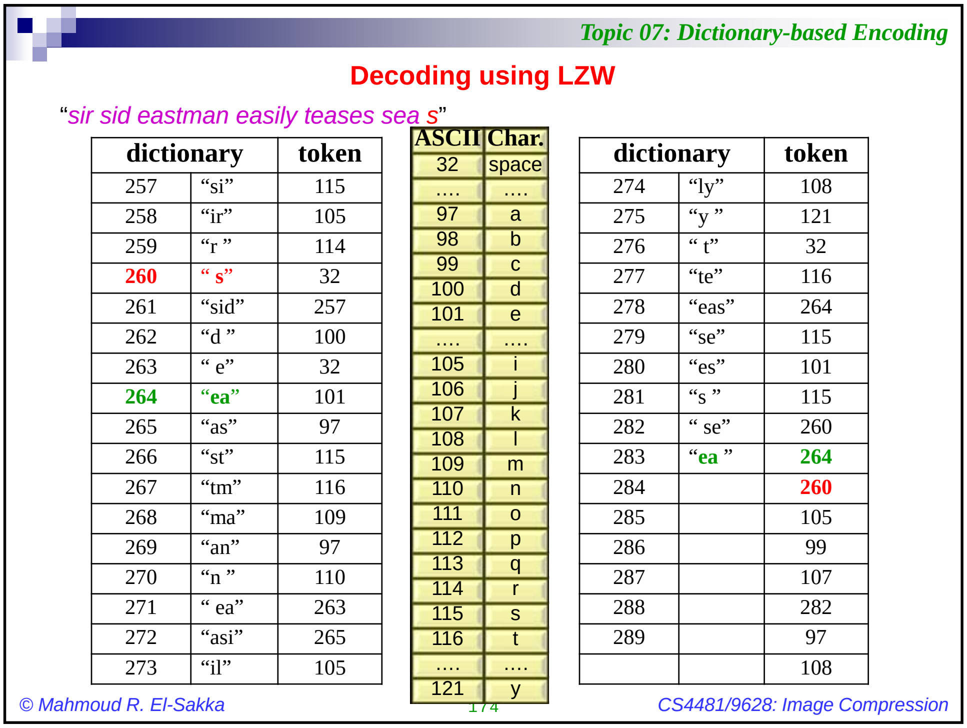
"*sir sid eastman easily teases sea*"

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	**" s"**	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	**"ea"**	101
265	"as"	97
266	"st"	115
267	"tm"	116
268	"ma"	109
269	"an"	97
270	"n "	110
271	" ea"	263
272	"asi"	265
273	"il"	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	"ly"	108
275	"y "	121
276	" t"	32
277	"te"	116
278	"eas"	264
279	"se"	115
280	"es"	101
281	"s "	115
282	" **se**"	**260**
283		**264**
284		260
285		105
286		99
287		107
288		282
289		97
		108

© Mahmoud R. El-Sakka 173 CS4481/9628: Image Compression



Topic 07: Dictionary-based Encoding

Decoding using LZW

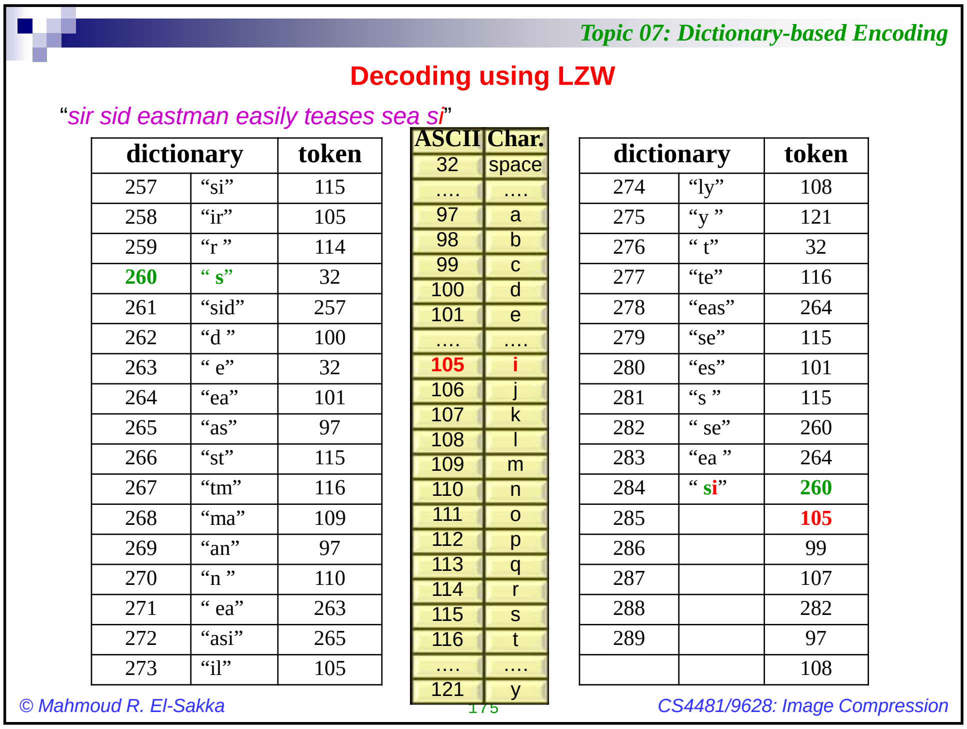
"*sir sid eastman easily teases sea s*"

dictionary		token
257	"si"	115
258	"ir"	105
259	"r "	114
260	**" s"**	32
261	"sid"	257
262	"d "	100
263	" e"	32
264	**"ea"**	101
265	"as"	97
266	"st"	115
267	"tm"	116
268	"ma"	109
269	"an"	97
270	"n "	110
271	" ea"	263
272	"asi"	265
273	"il"	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	"ly"	108
275	"y "	121
276	" t"	32
277	"te"	116
278	"eas"	264
279	"se"	115
280	"es"	101
281	"s "	115
282	" se"	260
283	"**ea** "	**264**
284		**260**
285		105
286		99
287		107
288		282
289		97
		108

© Mahmoud R. El-Sakka 174 CS4481/9628: Image Compression



Topic 07: Dictionary-based Encoding

Decoding using LZW

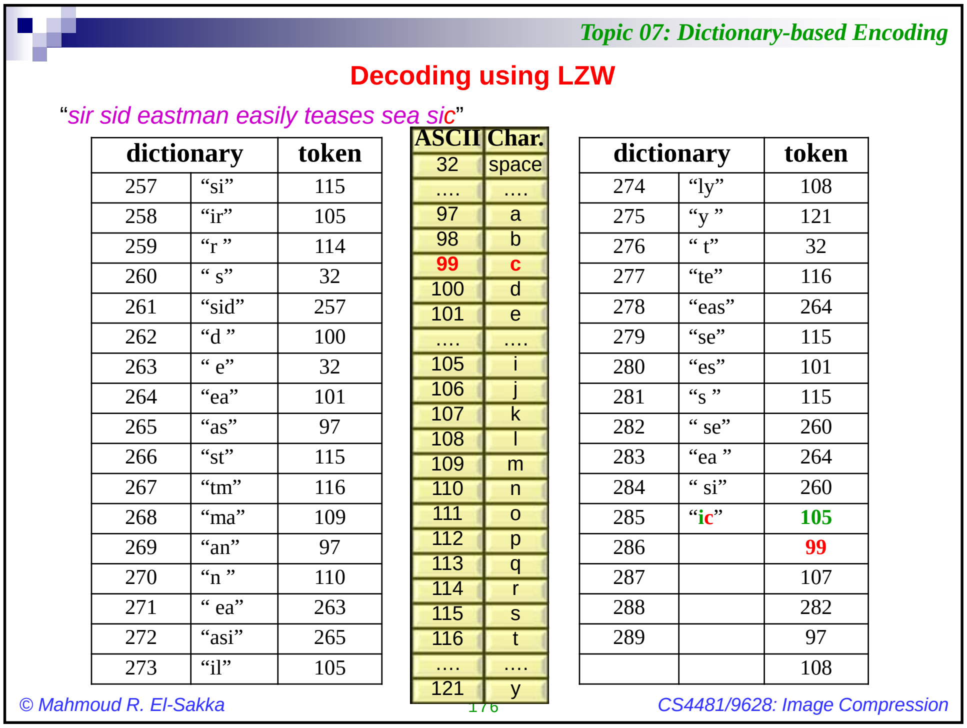
“*sir sid eastman easily teases sea si*”

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	**“ s”**	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“asi”	265
273	“il”	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	**i**
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	“ly”	108
275	“y ”	121
276	“ t”	32
277	“te”	116
278	“eas”	264
279	“se”	115
280	“es”	101
281	“s ”	115
282	“ se”	260
283	“ea ”	264
284	“ **si**”	**260**
285		**105**
286		99
287		107
288		282
289		97
		108

© Mahmoud R. El-Sakka 175 CS4481/9628: Image Compression



Topic 07: Dictionary-based Encoding

Decoding using LZW

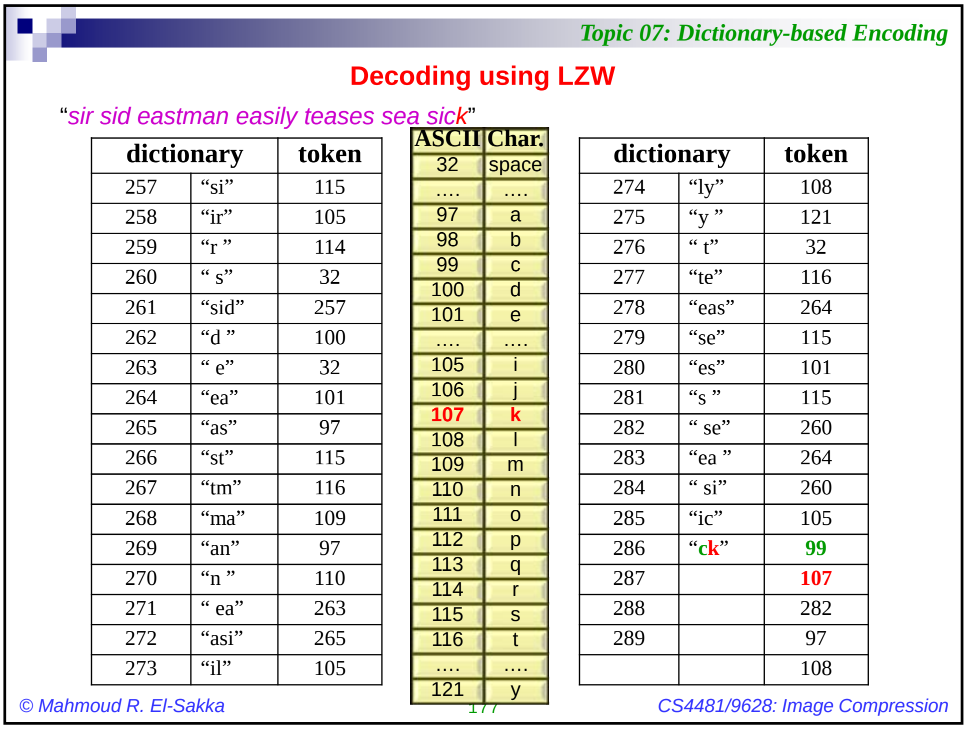
“*sir sid eastman easily teases sea sic*”

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“asi”	265
273	“il”	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	**c**
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	“ly”	108
275	“y ”	121
276	“ t”	32
277	“te”	116
278	“eas”	264
279	“se”	115
280	“es”	101
281	“s ”	115
282	“ se”	260
283	“ea ”	264
284	“ si”	260
285	“**ic**”	**105**
286		**99**
287		107
288		282
289		97
		108

© Mahmoud R. El-Sakka 176 CS4481/9628: Image Compression



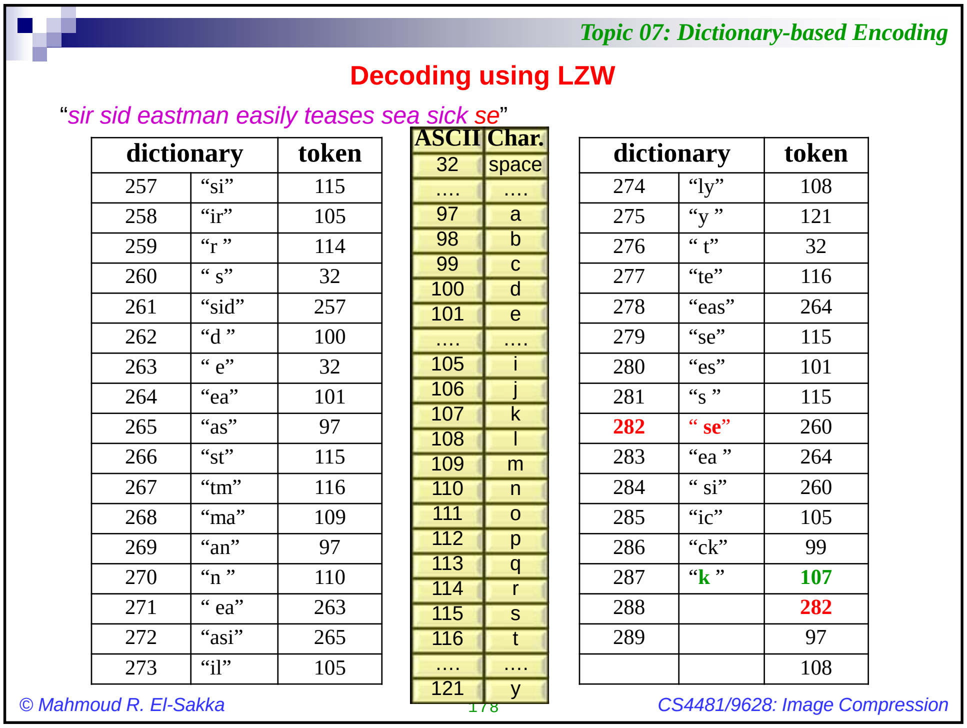
Decoding using LZW

“sir sid eastman easily teases sea sick”

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“asi”	265
273	“il”	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	“ly”	108
275	“y ”	121
276	“ t”	32
277	“te”	116
278	“eas”	264
279	“se”	115
280	“es”	101
281	“s ”	115
282	“ se”	260
283	“ea ”	264
284	“ si”	260
285	“ic”	105
286	“ck”	99
287		107
288		282
289		97
		108



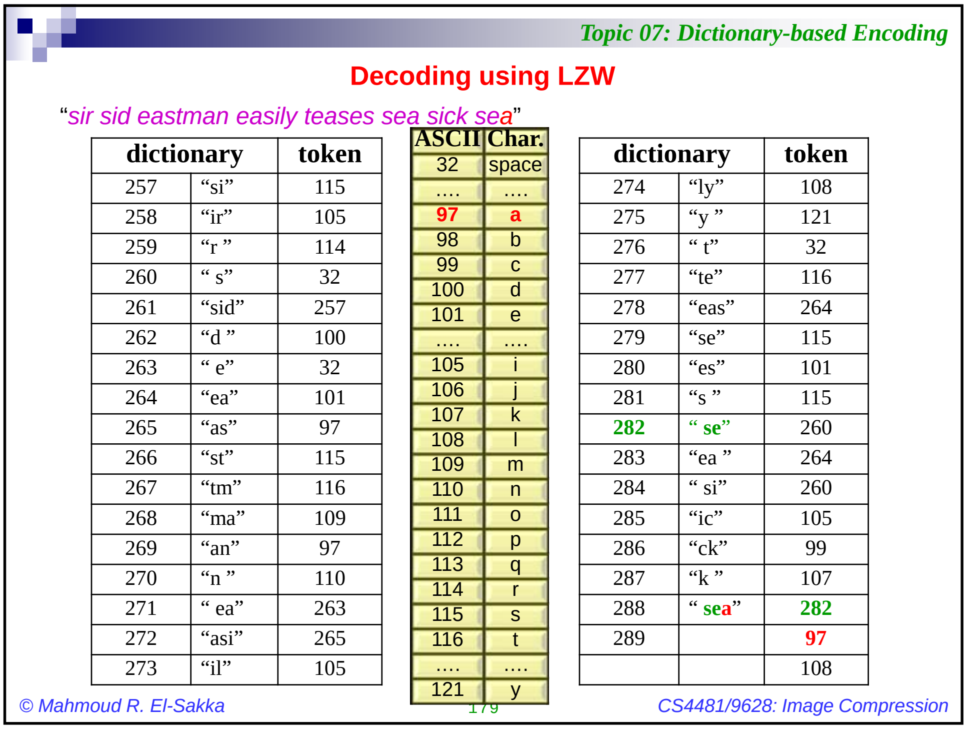
Decoding using LZW

“sir sid eastman easily teases sea sick se”

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“asi”	265
273	“il”	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	“ly”	108
275	“y ”	121
276	“ t”	32
277	“te”	116
278	“eas”	264
279	“se”	115
280	“es”	101
281	“s ”	115
282	“ se”	260
283	“ea ”	264
284	“ si”	260
285	“ic”	105
286	“ck”	99
287	“k ”	107
288		282
289		97
		108



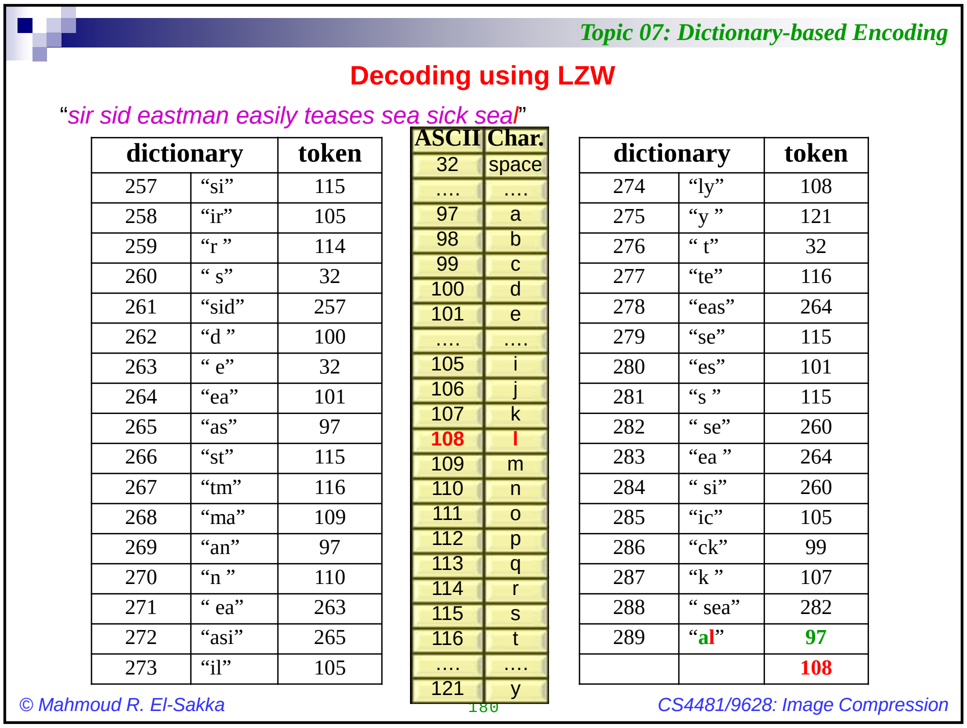
Decoding using LZW

“*sir sid eastman easily teases sea sick sea*”

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“asi”	265
273	“il”	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	“ly”	108
275	“y ”	121
276	“ t”	32
277	“te”	116
278	“eas”	264
279	“se”	115
280	“es”	101
281	“s ”	115
282	**“ se”**	260
283	“ea ”	264
284	“ si”	260
285	“ic”	105
286	“ck”	99
287	“k ”	107
288	“ **sea**”	**282**
289		**97**
		108



Decoding using LZW

“*sir sid eastman easily teases sea sick seal*”

dictionary		token
257	“si”	115
258	“ir”	105
259	“r ”	114
260	“ s”	32
261	“sid”	257
262	“d ”	100
263	“ e”	32
264	“ea”	101
265	“as”	97
266	“st”	115
267	“tm”	116
268	“ma”	109
269	“an”	97
270	“n ”	110
271	“ ea”	263
272	“asi”	265
273	“il”	105

ASCII	Char.
32	space
….	….
97	a
98	b
99	c
100	d
101	e
….	….
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
….	….
121	y

dictionary		token
274	“ly”	108
275	“y ”	121
276	“ t”	32
277	“te”	116
278	“eas”	264
279	“se”	115
280	“es”	101
281	“s ”	115
282	“ se”	260
283	“ea ”	264
284	“ si”	260
285	“ic”	105
286	“ck”	99
287	“k ”	107
288	“ sea”	282
289	“**al**”	**97**
		108

LZW

- The Unix **`compress`** command utilized LZW
  - □ Starts with a dictionary of size 512 entries, and hence *dictionary pointers* are represented by ***9 bits each***
  - □ Once the dictionary has filled up, the size of the dictionary is doubled to 1024 entries, and hence *dictionary pointers* are represented by ***10 bits each***
  - □ The size of the dictionary is progressively doubled as it fills up
  - □ The maximum *dictionary pointer* size can be set by the user, where this maximum size belongs to {9, 10, 11, 12, 13, 14, 15, 16}, with 16 bits being the default
  - □ Once the dictionary reaches its maximum capacity, the **`compress`** algorithm monitors the compression ratio
    - If the compression ratio falls below a certain threshold
      - □ The dictionary is flushed, using the dictionary entry number 256
      - □ The dictionary building process is initialized and re-started

      This way, the dictionary always reflects the local characteristics of the source

LZW

- In LZW, codeword encoding is not used;
  i.e., the compressed file will consists of a stream of bits representing the *dictionary pointers* sequence
- Since *dictionary pointers* sizes start by 9 bits and may increase, we need to do some bit-operations to efficiently bin-pack these 9 bits in bytes
- Let us consider the last example, where the generated *dictionary pointers* are 115, 105, 114, 32, …., 97, and 108
- All these *dictionary pointers* are 9 bits each
- In the next few slides, we will see how to bin-pack them in 8 bits

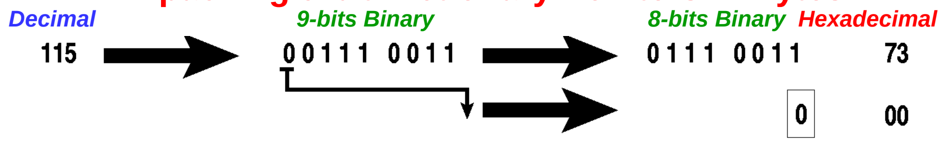
Bin-packing 9-bit Dictionary Pointers in Bytes

Decimal		9-bits Binary		8-bits Binary	Hexadecimal
115	→	00111 0011			

Bin-packing 9-bit Dictionary Pointers in Bytes

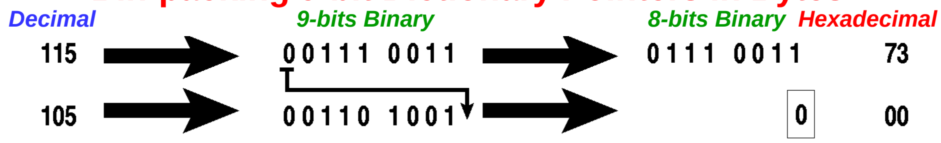
Decimal		9-bits Binary		8-bits Binary	Hexadecimal
115	→	$\underline{0}$0111 0011	→	0111 0011	73

Bin-packing 9-bit Dictionary Pointers in Bytes



Decimal	*9-bits Binary*	*8-bits Binary*	*Hexadecimal*
115	00111 0011	0111 0011	73
		$\boxed{0}$	00

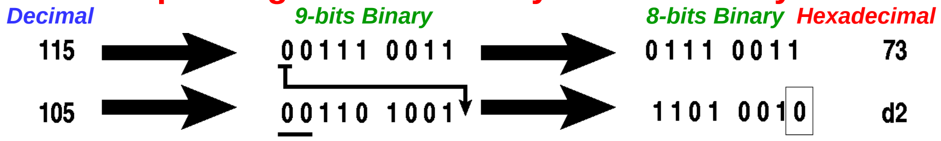
Bin-packing 9-bit Dictionary Pointers in Bytes



Decimal	*9-bits Binary*	*8-bits Binary*	*Hexadecimal*
115	00111 0011	0111 0011	73
105	00110 1001	$\boxed{0}$	00

Topic 07: Dictionary-based Encoding

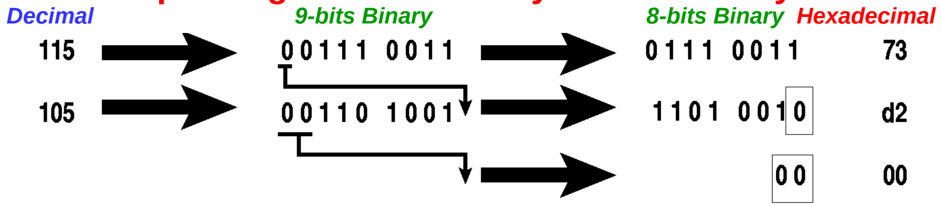
Bin-packing 9-bit Dictionary Pointers in Bytes



Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2

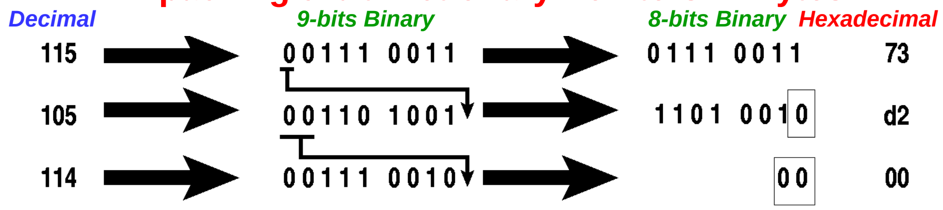
Topic 07: Dictionary-based Encoding

Bin-packing 9-bit Dictionary Pointers in Bytes



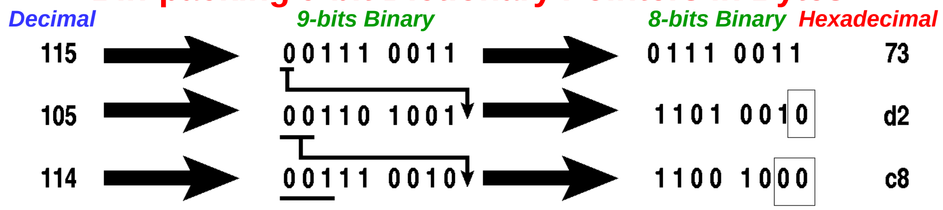
Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2
		00	00

Bin-packing 9-bit Dictionary Pointers in Bytes

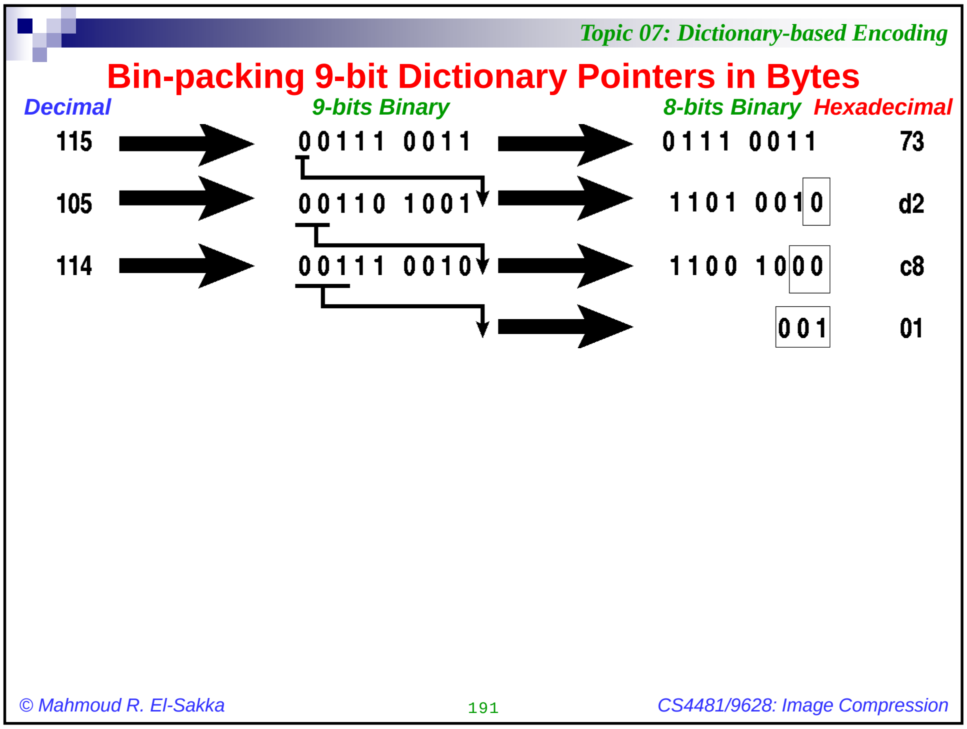


Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2
114	00111 0010	00	00

Bin-packing 9-bit Dictionary Pointers in Bytes

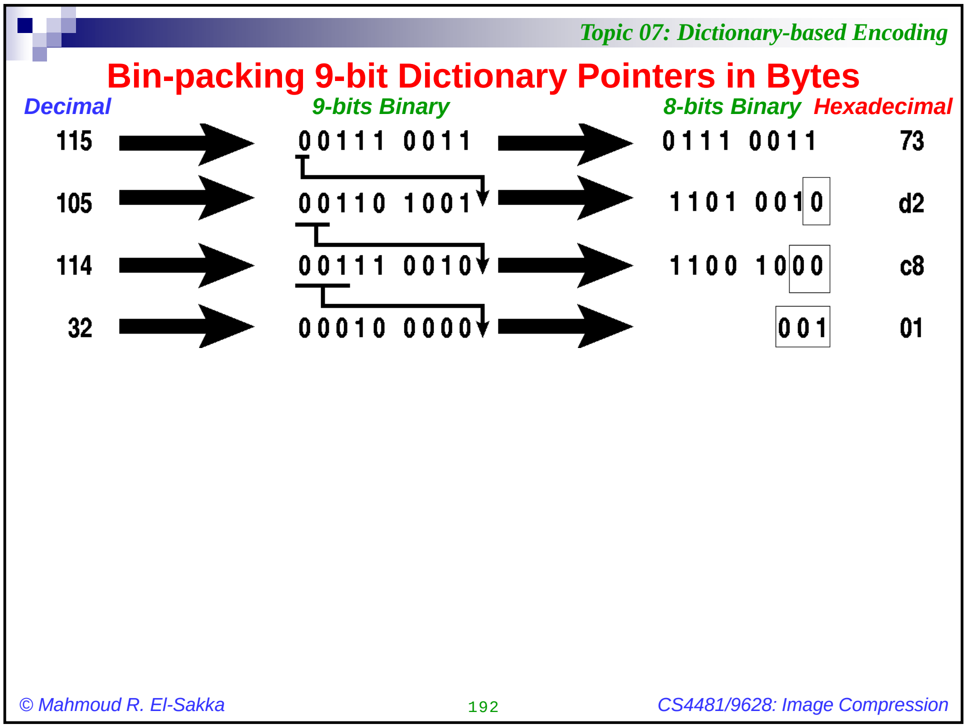


Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2
114	00111 0010	1100 1000	c8



Bin-packing 9-bit Dictionary Pointers in Bytes

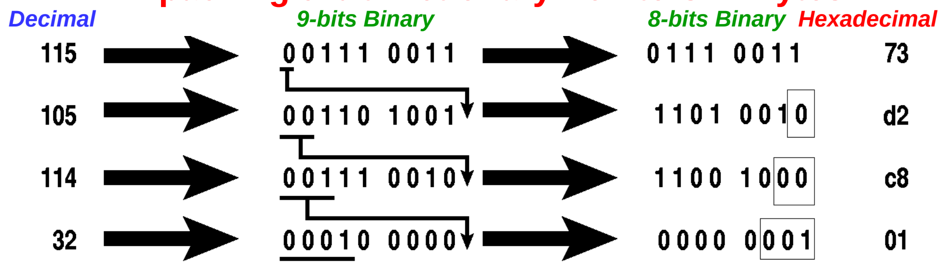
Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2
114	00111 0010	1100 1000	c8
		001	01



Bin-packing 9-bit Dictionary Pointers in Bytes

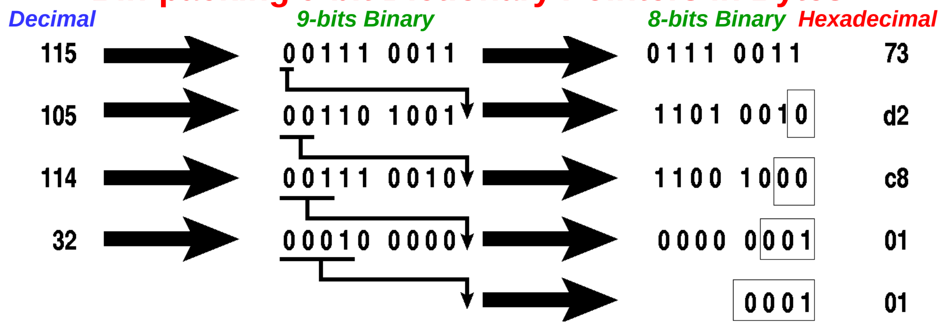
Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2
114	00111 0010	1100 1000	c8
32	00010 0000	001	01

Bin-packing 9-bit Dictionary Pointers in Bytes

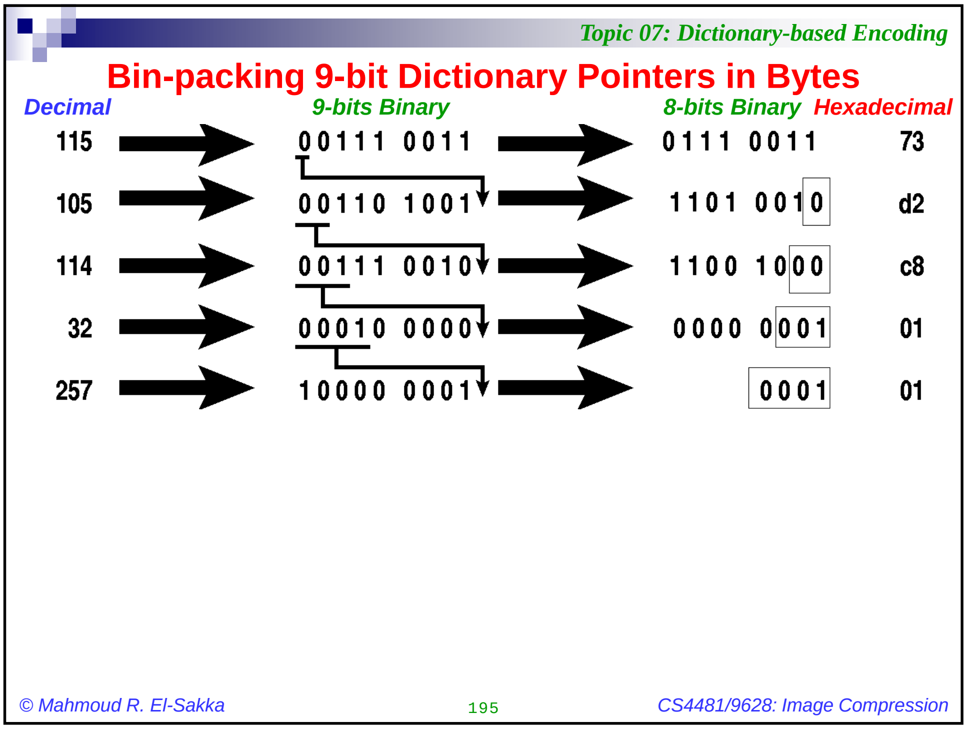


Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2
114	00111 0010	1100 1000	c8
32	00010 0000	0000 0001	01

Bin-packing 9-bit Dictionary Pointers in Bytes

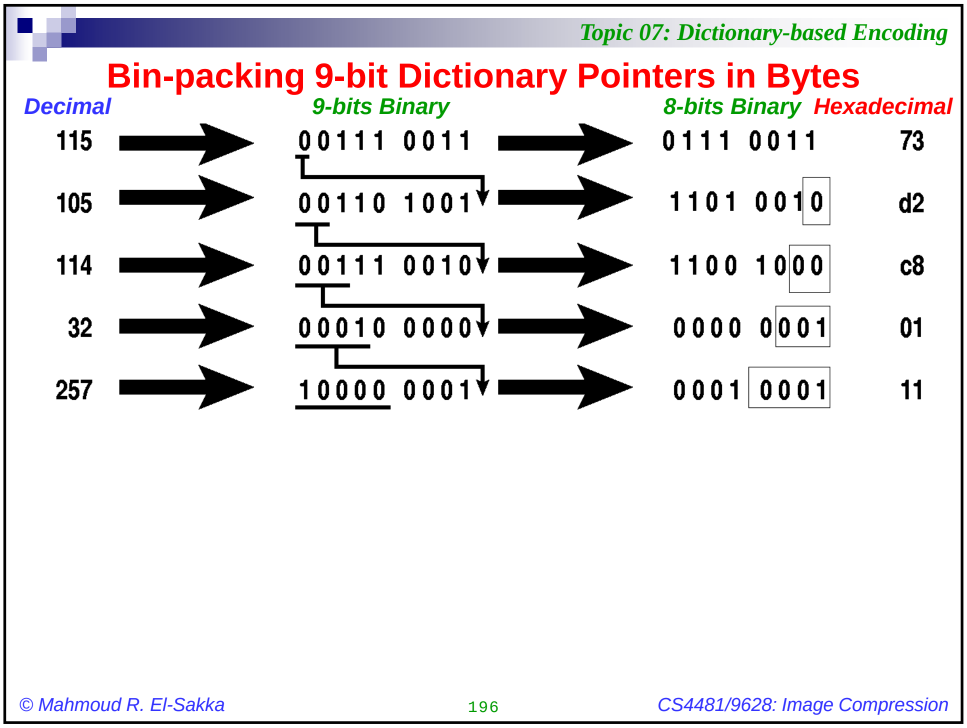


Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2
114	00111 0010	1100 1000	c8
32	00010 0000	0000 0001	01
		0001	01



Bin-packing 9-bit Dictionary Pointers in Bytes

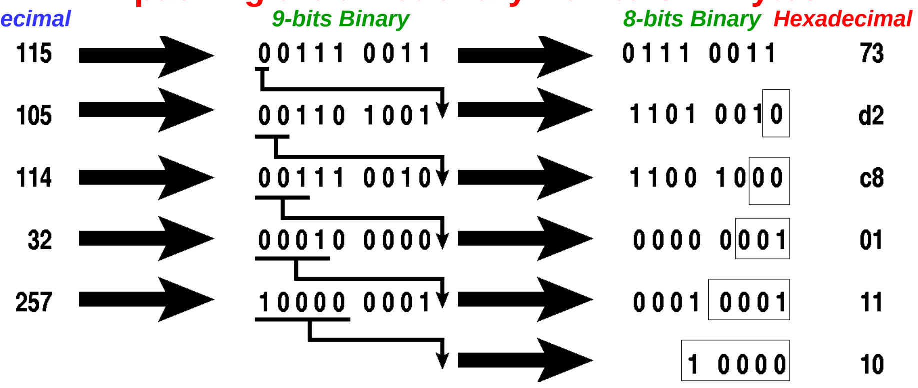
Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2
114	00111 0010	1100 1000	c8
32	00010 0000	0000 0001	01
257	10000 0001	0001	01



Bin-packing 9-bit Dictionary Pointers in Bytes

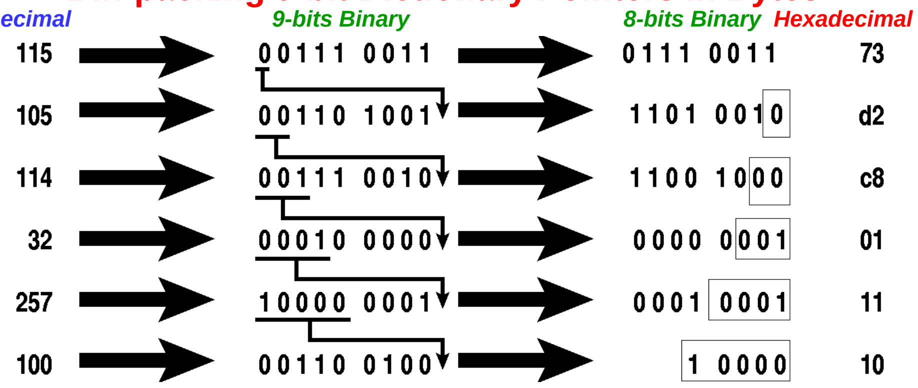
Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2
114	00111 0010	1100 1000	c8
32	00010 0000	0000 0001	01
257	10000 0001	0001 0001	11

Bin-packing 9-bit Dictionary Pointers in Bytes



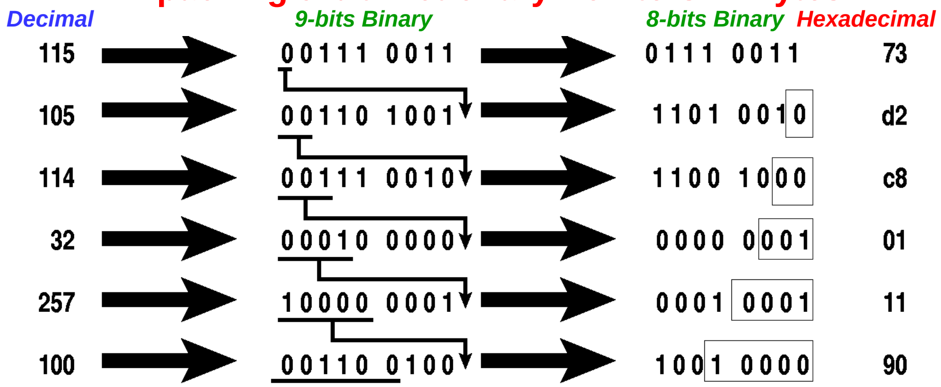
Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2
114	00111 0010	1100 1000	c8
32	00010 0000	0000 0001	01
257	10000 0001	0001 0001	11
		1 0000	10

Bin-packing 9-bit Dictionary Pointers in Bytes



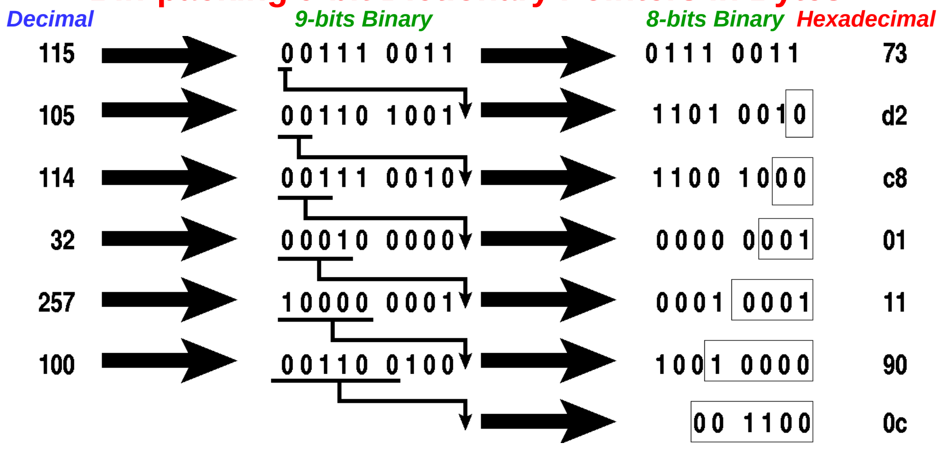
Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2
114	00111 0010	1100 1000	c8
32	00010 0000	0000 0001	01
257	10000 0001	0001 0001	11
100	00110 0100	1 0000	10

Bin-packing 9-bit Dictionary Pointers in Bytes

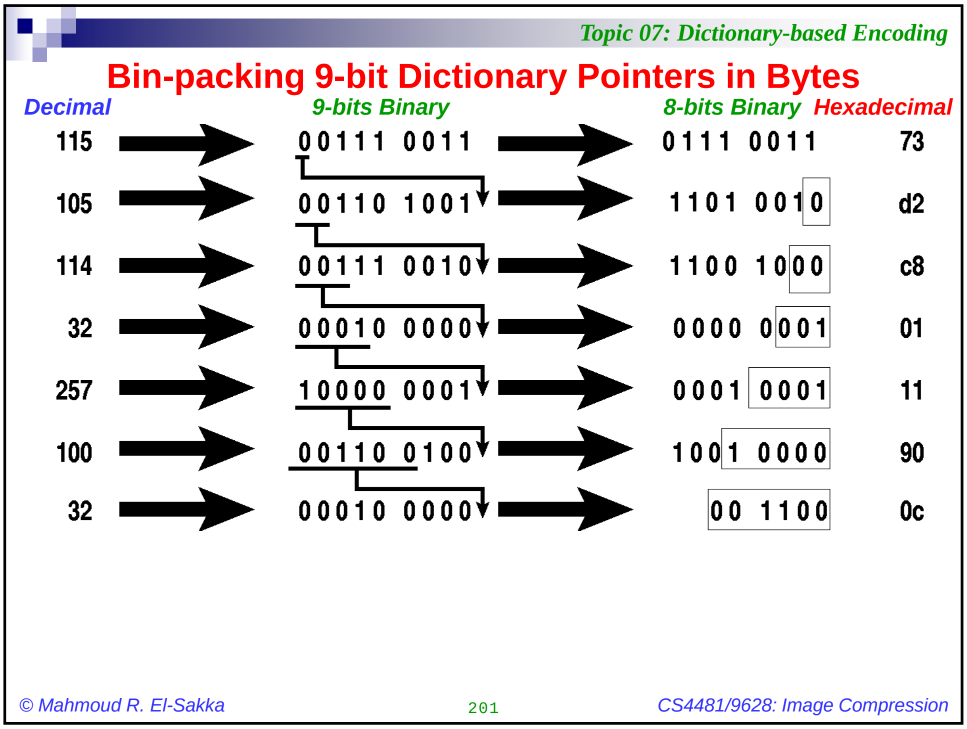


Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2
114	00111 0010	1100 1000	c8
32	00010 0000	0000 0001	01
257	10000 0001	0001 0001	11
100	00110 0100	1001 0000	90

Bin-packing 9-bit Dictionary Pointers in Bytes

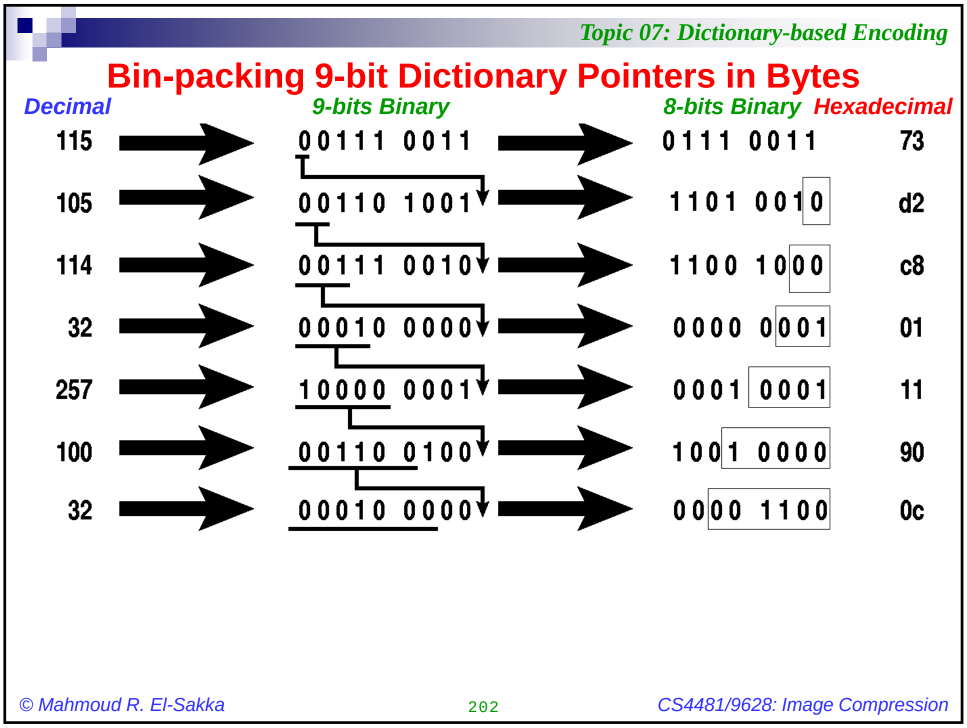


Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2
114	00111 0010	1100 1000	c8
32	00010 0000	0000 0001	01
257	10000 0001	0001 0001	11
100	00110 0100	1001 0000	90
		00 1100	0c



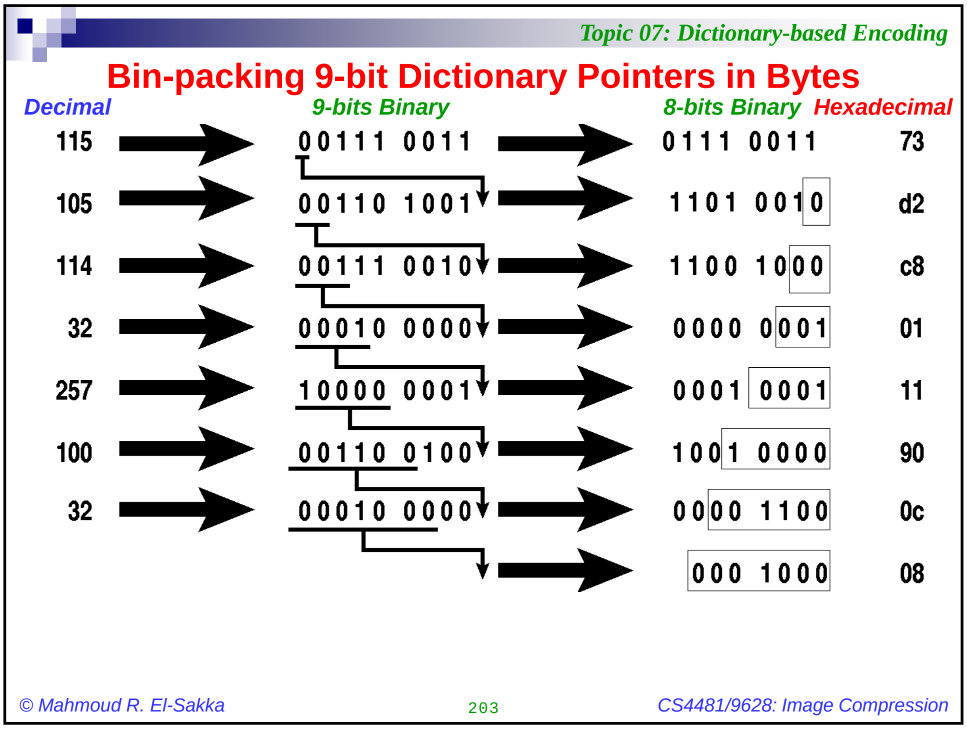
Bin-packing 9-bit Dictionary Pointers in Bytes

Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2
114	00111 0010	1100 1000	c8
32	00010 0000	0000 0001	01
257	10000 0001	0001 0001	11
100	00110 0100	1001 0000	90
32	00010 0000	00 1100	0c



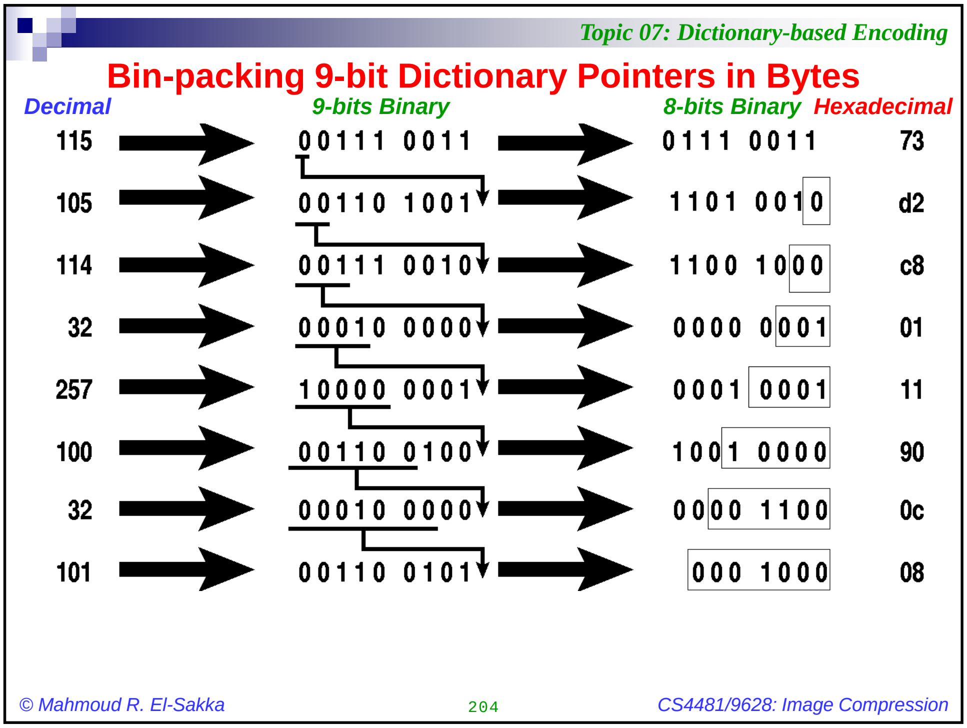
Bin-packing 9-bit Dictionary Pointers in Bytes

Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2
114	00111 0010	1100 1000	c8
32	00010 0000	0000 0001	01
257	10000 0001	0001 0001	11
100	00110 0100	1001 0000	90
32	00010 0000	0000 1100	0c



Bin-packing 9-bit Dictionary Pointers in Bytes

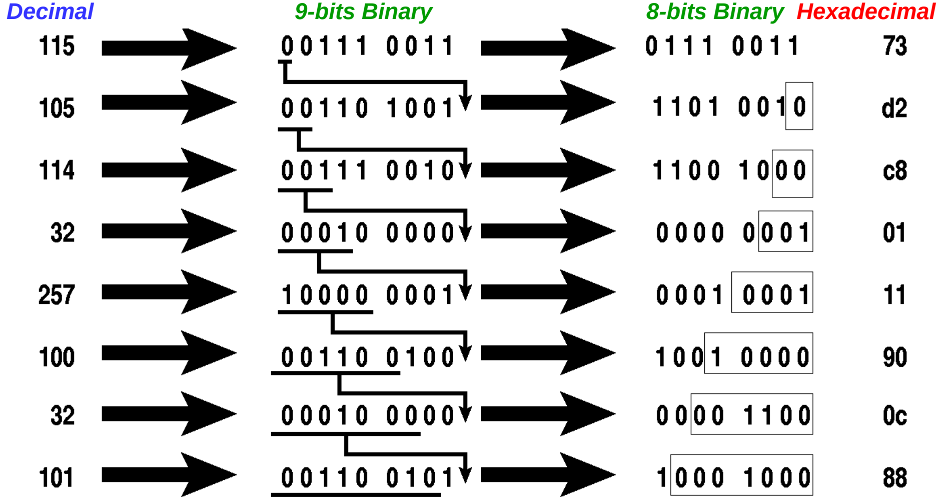
Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2
114	00111 0010	1100 1000	c8
32	00010 0000	0000 0001	01
257	10000 0001	0001 0001	11
100	00110 0100	1001 0000	90
32	00010 0000	0000 1100	0c
		000 1000	08



Bin-packing 9-bit Dictionary Pointers in Bytes

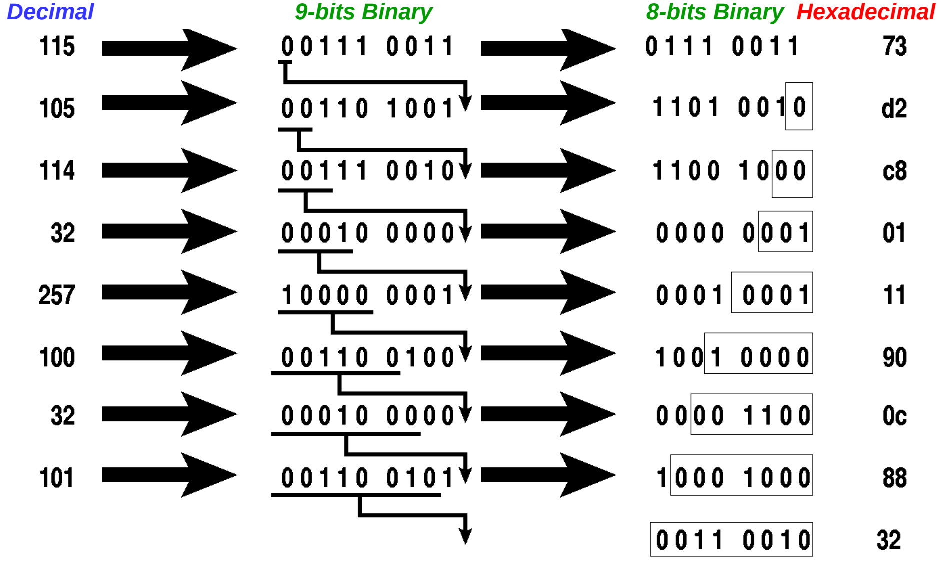
Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2
114	00111 0010	1100 1000	c8
32	00010 0000	0000 0001	01
257	10000 0001	0001 0001	11
100	00110 0100	1001 0000	90
32	00010 0000	0000 1100	0c
101	00110 0101	000 1000	08

Bin-packing 9-bit Dictionary Pointers in Bytes

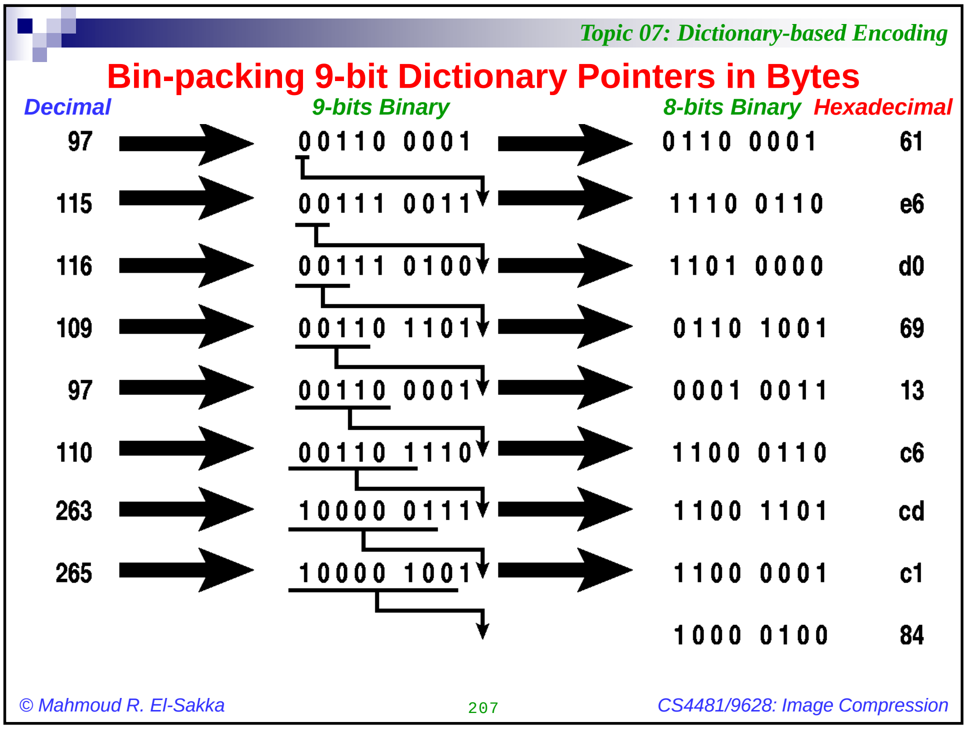


Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2
114	00111 0010	1100 1000	c8
32	00010 0000	0000 0001	01
257	10000 0001	0001 0001	11
100	00110 0100	1001 0000	90
32	00010 0000	0000 1100	0c
101	00110 0101	1000 1000	88

Bin-packing 9-bit Dictionary Pointers in Bytes

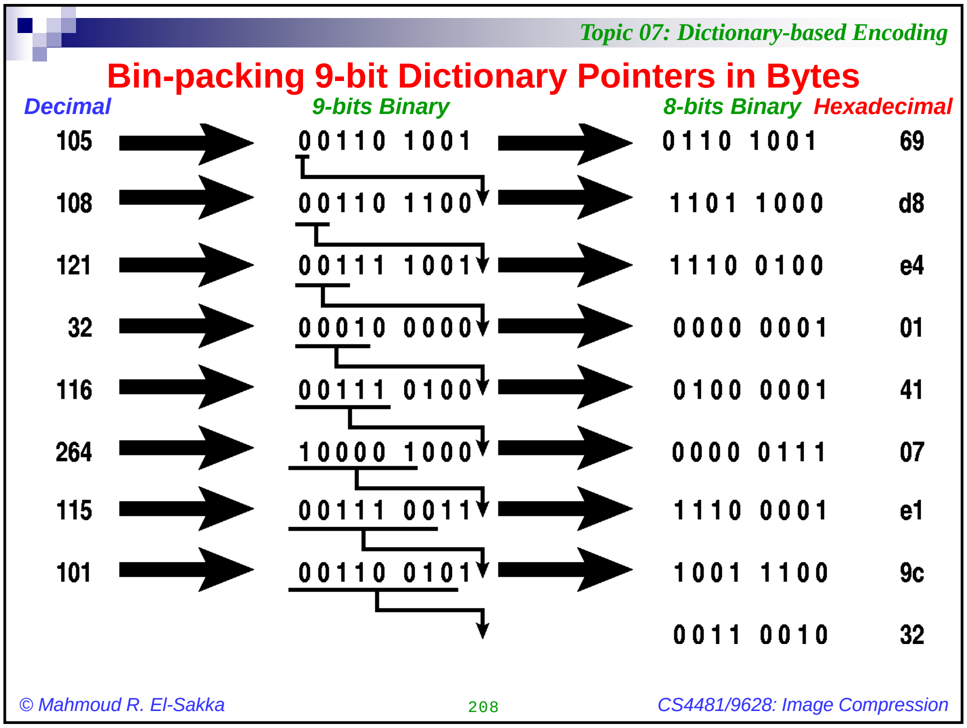


Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
105	00110 1001	1101 0010	d2
114	00111 0010	1100 1000	c8
32	00010 0000	0000 0001	01
257	10000 0001	0001 0001	11
100	00110 0100	1001 0000	90
32	00010 0000	0000 1100	0c
101	00110 0101	1000 1000	88
		0011 0010	32



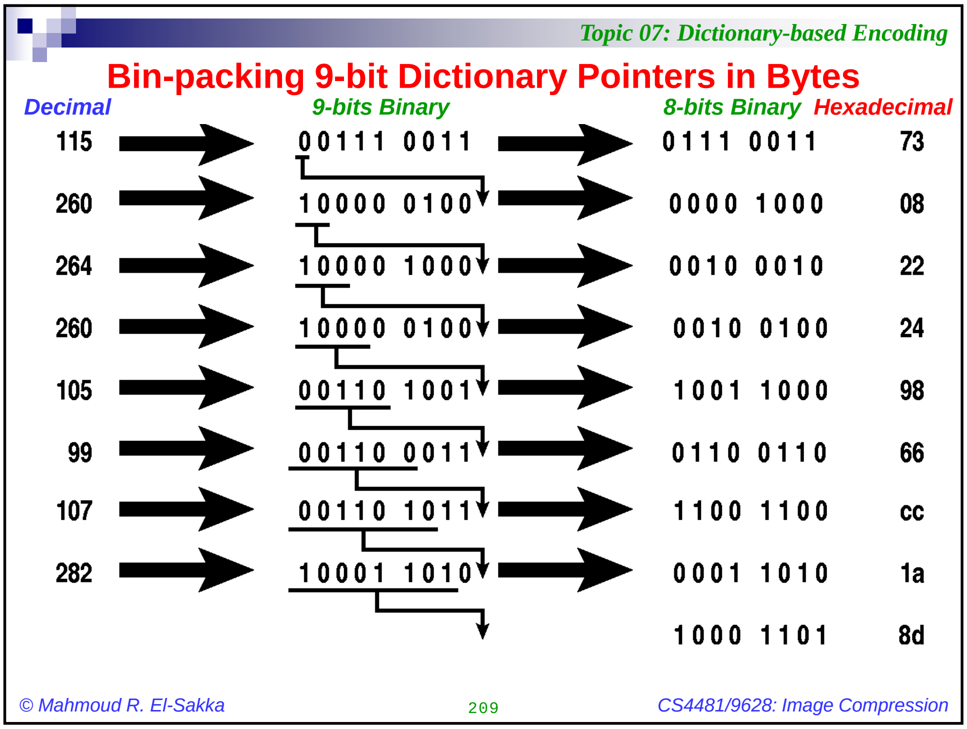
Bin-packing 9-bit Dictionary Pointers in Bytes

Decimal	9-bits Binary	8-bits Binary	Hexadecimal
97	00110 0001	0110 0001	61
115	00111 0011	1110 0110	e6
116	00111 0100	1101 0000	d0
109	00110 1101	0110 1001	69
97	00110 0001	0001 0011	13
110	00110 1110	1100 0110	c6
263	10000 0111	1100 1101	cd
265	10000 1001	1100 0001	c1
		1000 0100	84



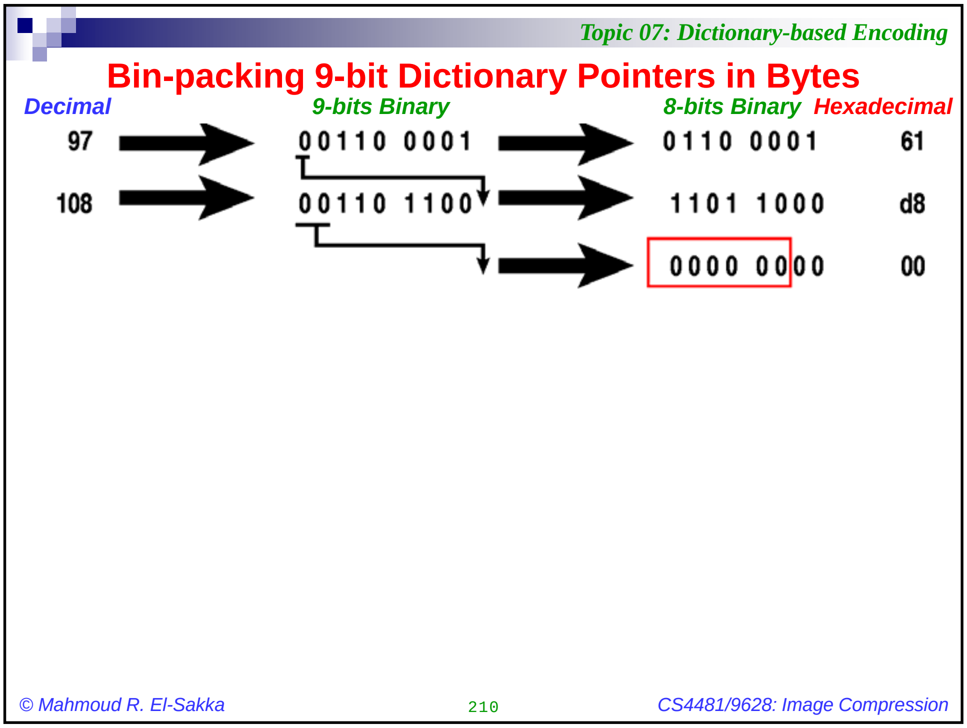
Bin-packing 9-bit Dictionary Pointers in Bytes

Decimal	9-bits Binary	8-bits Binary	Hexadecimal
105	00110 1001	0110 1001	69
108	00110 1100	1101 1000	d8
121	00111 1001	1110 0100	e4
32	00010 0000	0000 0001	01
116	00111 0100	0100 0001	41
264	10000 1000	0000 0111	07
115	00111 0011	1110 0001	e1
101	00110 0101	1001 1100	9c
		0011 0010	32



Bin-packing 9-bit Dictionary Pointers in Bytes

Decimal	9-bits Binary	8-bits Binary	Hexadecimal
115	00111 0011	0111 0011	73
260	10000 0100	0000 1000	08
264	10000 1000	0010 0010	22
260	10000 0100	0010 0100	24
105	00110 1001	1001 1000	98
99	00110 0011	0110 0110	66
107	00110 1011	1100 1100	cc
282	10001 1010	0001 1010	1a
		1000 1101	8d



Bin-packing 9-bit Dictionary Pointers in Bytes

Decimal	9-bits Binary	8-bits Binary	Hexadecimal
97	00110 0001	0110 0001	61
108	00110 1100	1101 1000	d8
		0000 0000	00

LZW

- The 9-bit sequence (115, 105, 114, 32, …., 97, and 108) is converted to 8-bit sequence (73, d2, c8, 01, …, 61, d8, and 00)
- This 8-bit sequence is the compressed LZW file

LZW

- The output of the Unix `compress` for a file that contains

  "sir sid eastman easily teases sea sick seal"

  can be seen by the **"`od -t x1 file_name.Z`"** command

```
0000000 1f 9d 90 73 d2 c8 01 11 90 0c 88 32 61 e6 d0 69
0000020 13 c6 cd c1 84 69 d8 e4 01 41 07 e1 9c 32 73 08
0000040 22 24 98 66 cc 1a 8d 61 d8 28 00
```

- Note that:
  - □ The pointer indices start from the fourth byte
  - □ The first two bytes ***(1f 9d)*** are the *Magic Number* for the Unix `compress`
  - □ The 5 least significant bits in the third byte ***(1001 0000)***, are used to identify the maximum length of the *dictionary pointers*; 16 in this example
  - □ **Why do we have "*28*" at the last line of the "`od`" output?**
  - □ You should use **"`-f`"** when compressing the file to force the command to produce output, even if your file is expanded

LZW

- To decompress an LZW file, we should convert the 8-bit sequence (73, d2, c8, 01, …, 61, d8 and 00) into 9-bit dictionary pointers sequence (115, 105, 114, 32, …., 97, and 108)
- In the next few slides, we will see how to unpack these bytes into 9-bit dictionary pointers sequence

Unpacking Bytes into 9-bit Dictionary Pointers

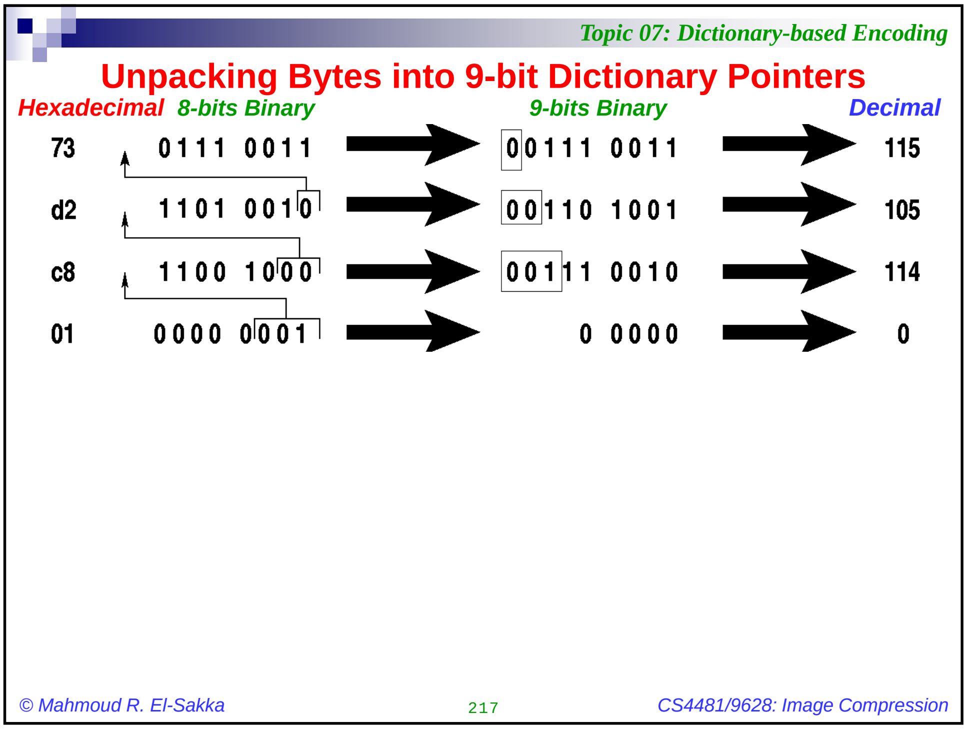
Hexadecimal	8-bits Binary		9-bits Binary		Decimal
73	0111 0011	→	0111 0011	→	115

Unpacking Bytes into 9-bit Dictionary Pointers

Hexadecimal	8-bits Binary	9-bits Binary	Decimal
73	0111 0011	$\boxed{0}$0111 0011	115
d2	1101 001$\boxed{0}$	110 1001	105

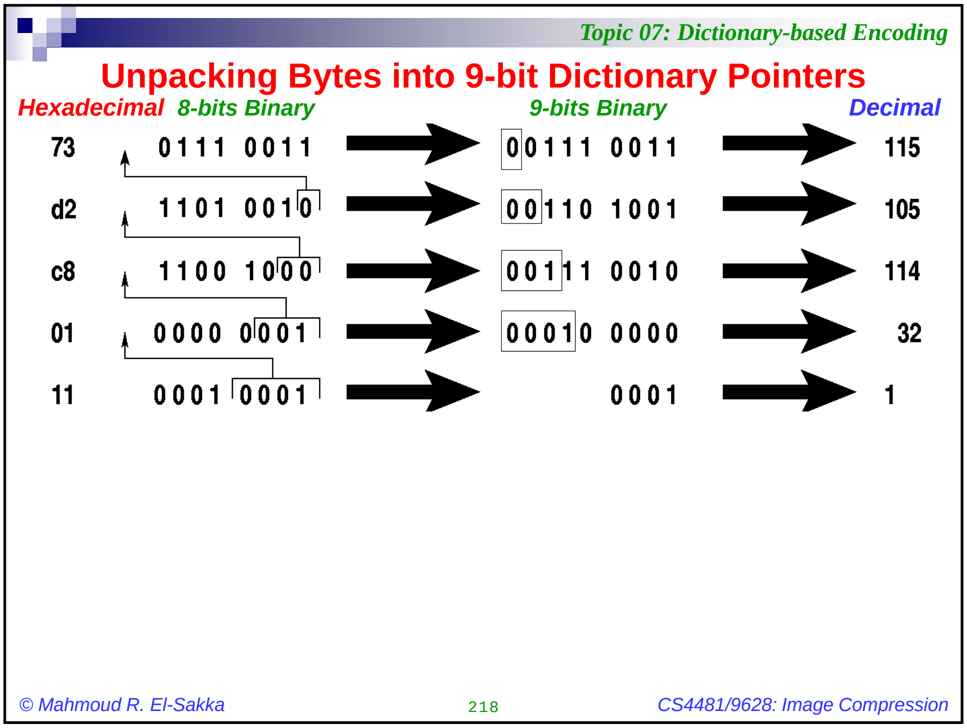
Unpacking Bytes into 9-bit Dictionary Pointers

Hexadecimal	8-bits Binary	9-bits Binary	Decimal
73	0111 0011	$\boxed{0}$0111 0011	115
d2	1101 001$\boxed{0}$	$\boxed{00}$110 1001	105
c8	1100 10$\boxed{00}$	11 0010	50



Unpacking Bytes into 9-bit Dictionary Pointers

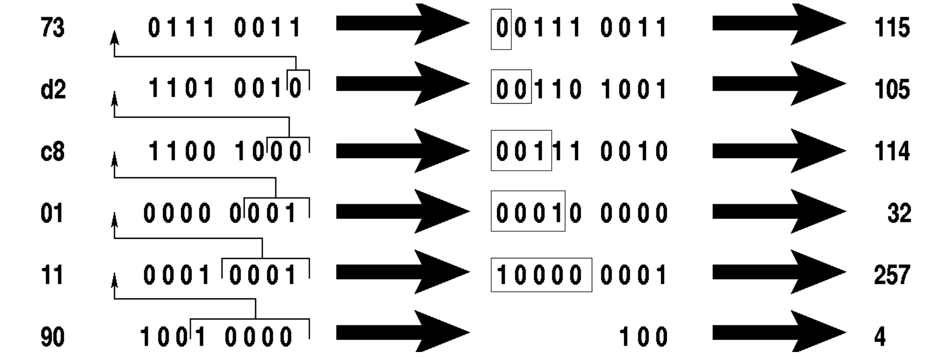
Hexadecimal	8-bits Binary		9-bits Binary		Decimal
73	0111 0011	→	00111 0011	→	115
d2	1101 0010	→	00110 1001	→	105
c8	1100 1000	→	00111 0010	→	114
01	0000 0001	→	0 0000	→	0



Unpacking Bytes into 9-bit Dictionary Pointers

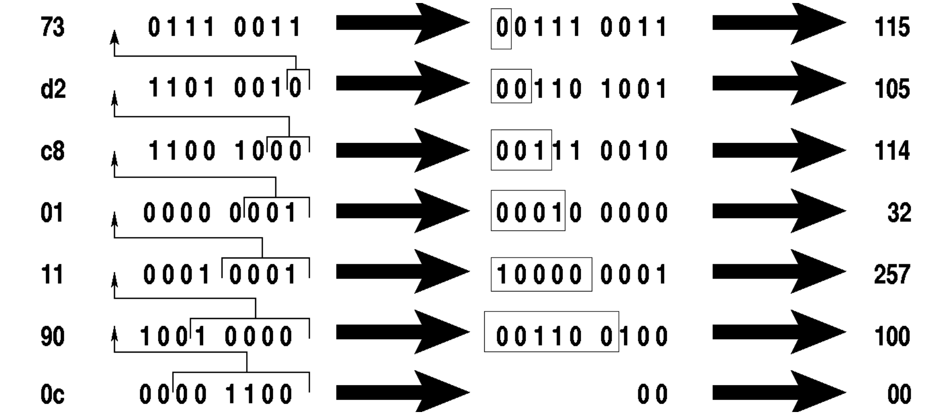
Hexadecimal	8-bits Binary		9-bits Binary		Decimal
73	0111 0011	→	00111 0011	→	115
d2	1101 0010	→	00110 1001	→	105
c8	1100 1000	→	00111 0010	→	114
01	0000 0001	→	00010 0000	→	32
11	0001 0001	→	0001	→	1

Unpacking Bytes into 9-bit Dictionary Pointers



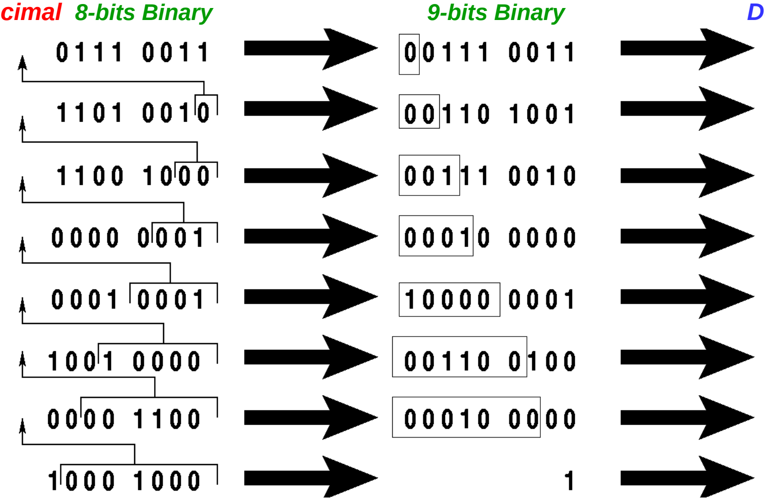
Hexadecimal	8-bits Binary	9-bits Binary	Decimal
73	0111 0011	00111 0011	115
d2	1101 0010	00110 1001	105
c8	1100 1000	00111 0010	114
01	0000 0001	00010 0000	32
11	0001 0001	10000 0001	257
90	1001 0000	100	4

Unpacking Bytes into 9-bit Dictionary Pointers



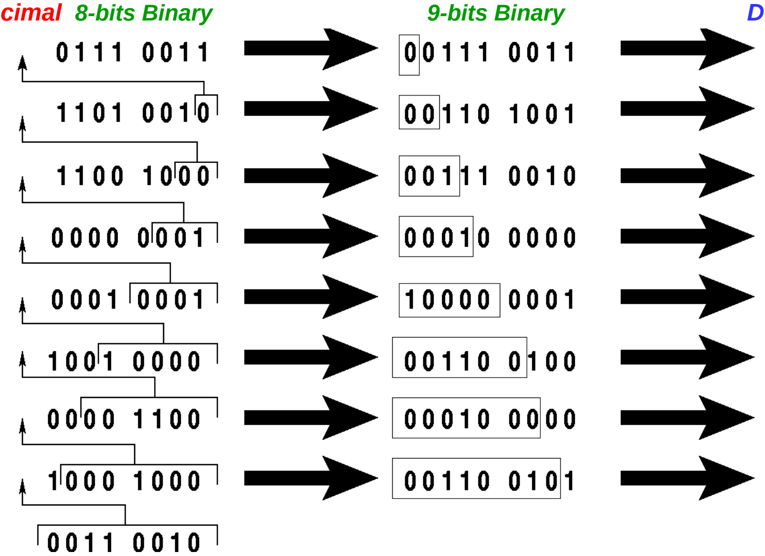
Hexadecimal	8-bits Binary	9-bits Binary	Decimal
73	0111 0011	00111 0011	115
d2	1101 0010	00110 1001	105
c8	1100 1000	00111 0010	114
01	0000 0001	00010 0000	32
11	0001 0001	10000 0001	257
90	1001 0000	00110 0100	100
0c	0000 1100	00	00

Unpacking Bytes into 9-bit Dictionary Pointers

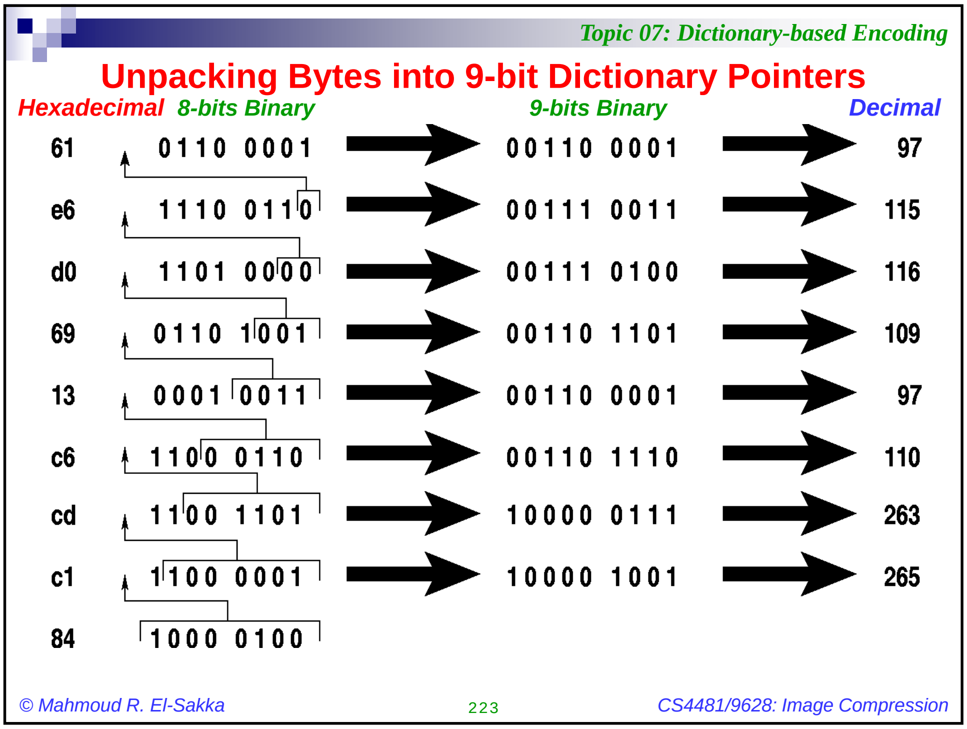


Hexadecimal	8-bits Binary	9-bits Binary	Decimal
73	0111 0011	00111 0011	115
d2	1101 0010	00110 1001	105
c8	1100 1000	00111 0010	114
01	0000 0001	00010 0000	32
11	0001 0001	10000 0001	257
90	1001 0000	00110 0100	100
0c	0000 1100	00010 0000	32
88	1000 1000	1	1

Unpacking Bytes into 9-bit Dictionary Pointers

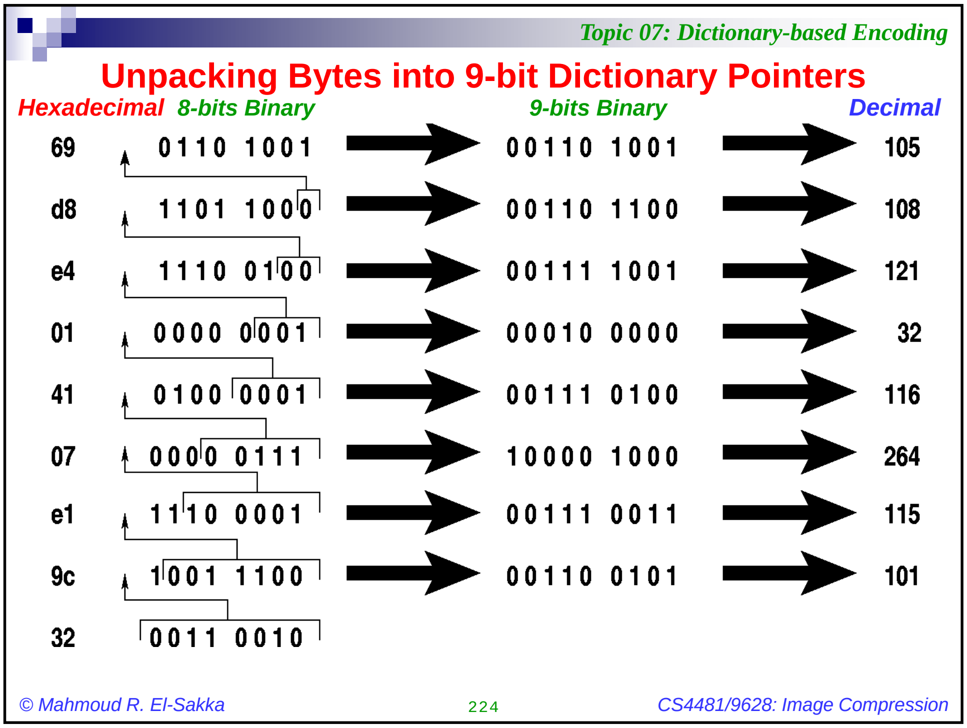


Hexadecimal	8-bits Binary	9-bits Binary	Decimal
73	0111 0011	00111 0011	115
d2	1101 0010	00110 1001	105
c8	1100 1000	00111 0010	114
01	0000 0001	00010 0000	32
11	0001 0001	10000 0001	257
90	1001 0000	00110 0100	100
0c	0000 1100	00010 0000	32
88	1000 1000	00110 0101	101
32	0011 0010		



Unpacking Bytes into 9-bit Dictionary Pointers

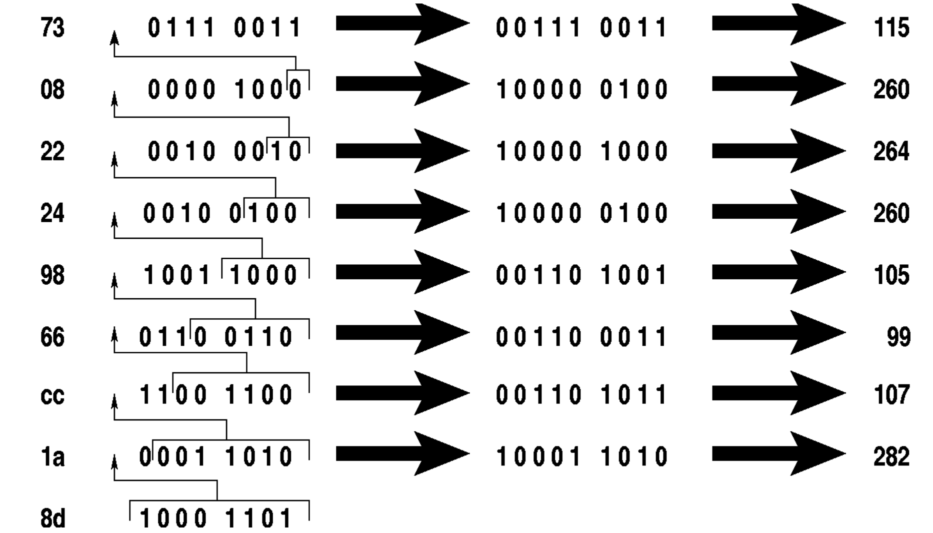
Hexadecimal	8-bits Binary	9-bits Binary	Decimal
61	0110 0001	00110 0001	97
e6	1110 0110	00111 0011	115
d0	1101 0000	00111 0100	116
69	0110 1001	00110 1101	109
13	0001 0011	00110 0001	97
c6	1100 0110	00110 1110	110
cd	1100 1101	10000 0111	263
c1	1100 0001	10000 1001	265
84	1000 0100		



Unpacking Bytes into 9-bit Dictionary Pointers

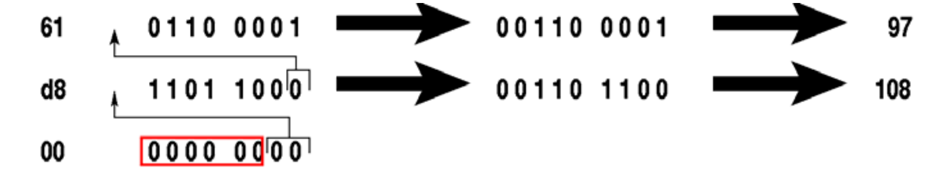
Hexadecimal	8-bits Binary	9-bits Binary	Decimal
69	0110 1001	00110 1001	105
d8	1101 1000	00110 1100	108
e4	1110 0100	00111 1001	121
01	0000 0001	00010 0000	32
41	0100 0001	00111 0100	116
07	0000 0111	10000 1000	264
e1	1110 0001	00111 0011	115
9c	1001 1100	00110 0101	101
32	0011 0010		

Unpacking Bytes into 9-bit Dictionary Pointers



Hexadecimal	8-bits Binary	9-bits Binary	Decimal
73	0111 0011	00111 0011	115
08	0000 1000	10000 0100	260
22	0010 0010	10000 1000	264
24	0010 0100	10000 0100	260
98	1001 1000	00110 1001	105
66	0110 0110	00110 0011	99
cc	1100 1100	00110 1011	107
1a	0001 1010	10001 1010	282
8d	1000 1101		

Unpacking Bytes into 9-bit Dictionary Pointers



Hexadecimal	8-bits Binary	9-bits Binary	Decimal
61	0110 0001	00110 0001	97
d8	1101 1000	00110 1100	108
00	0000 0000		

LZW

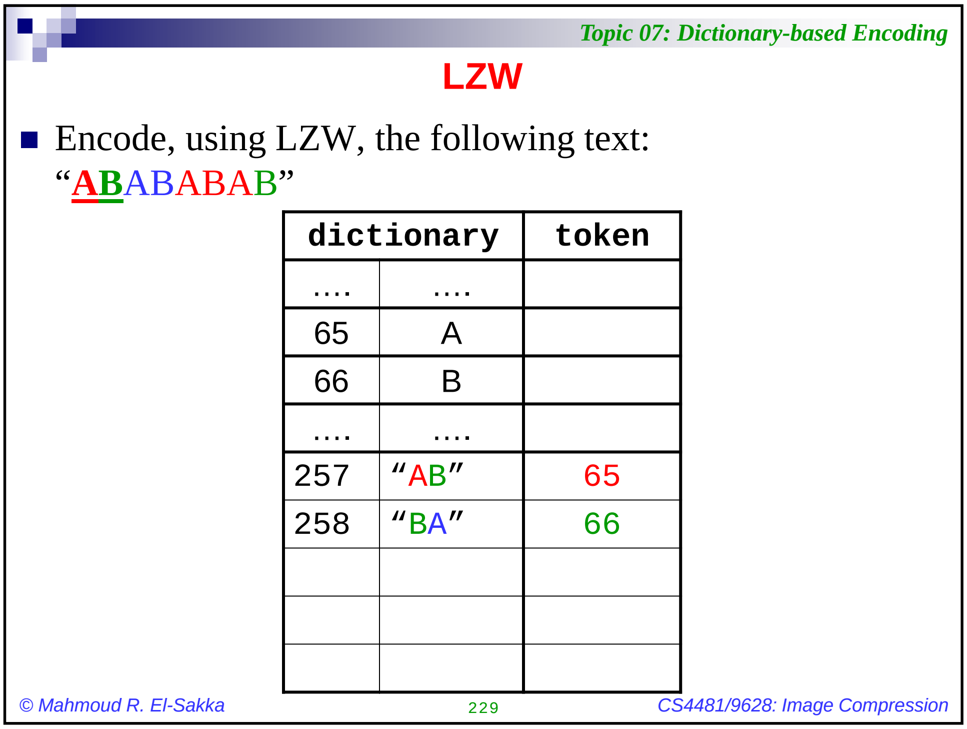
- Encode, using LZW, the following text:
“ABABABAB”

dictionary		token
….	….	
65	A	
66	B	
….	….	

LZW

- Encode, using LZW, the following text:
“ABABABAB”

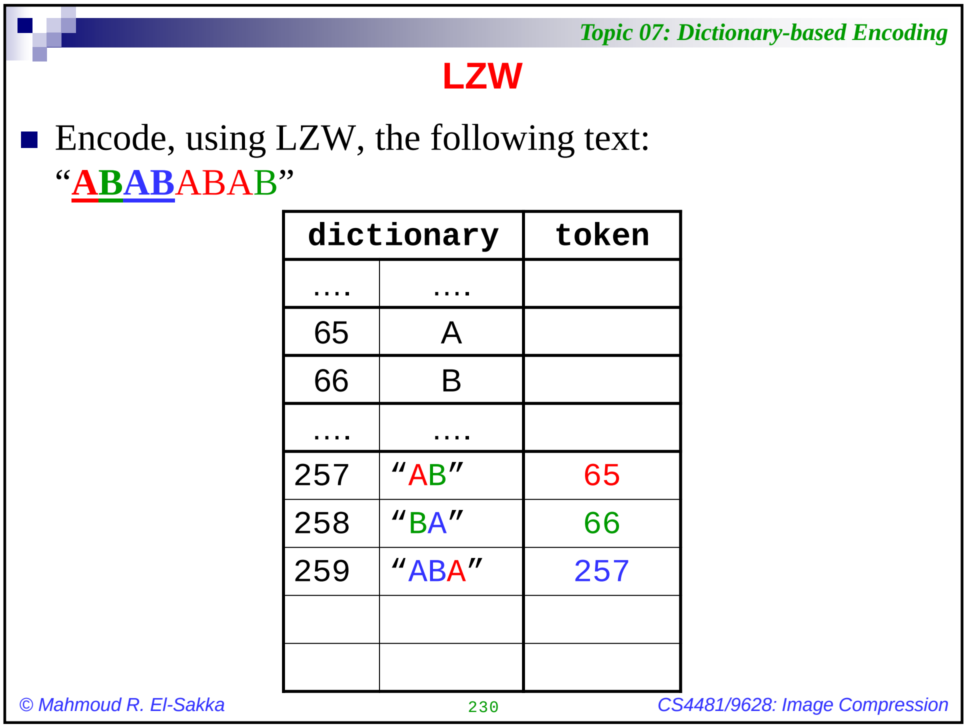
dictionary		token
….	….	
65	A	
66	B	
….	….	
257	“AB”	65



LZW

- Encode, using LZW, the following text: "ABABABAB"

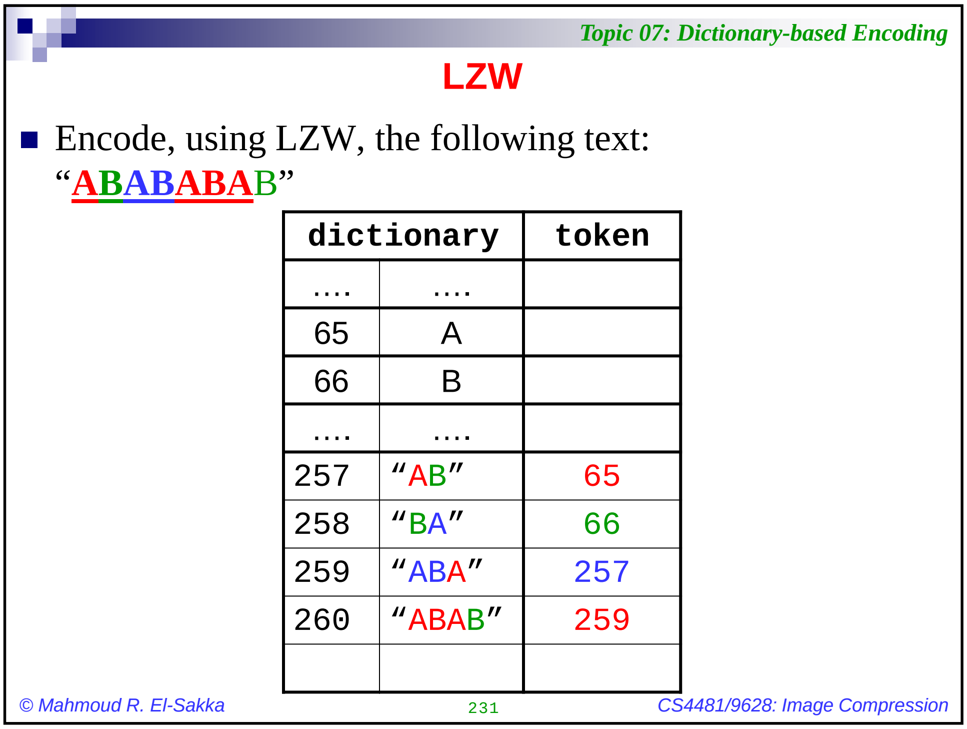
dictionary		token
….	….	
65	A	
66	B	
….	….	
257	"AB"	65
258	"BA"	66



LZW

- Encode, using LZW, the following text: "ABABABAB"

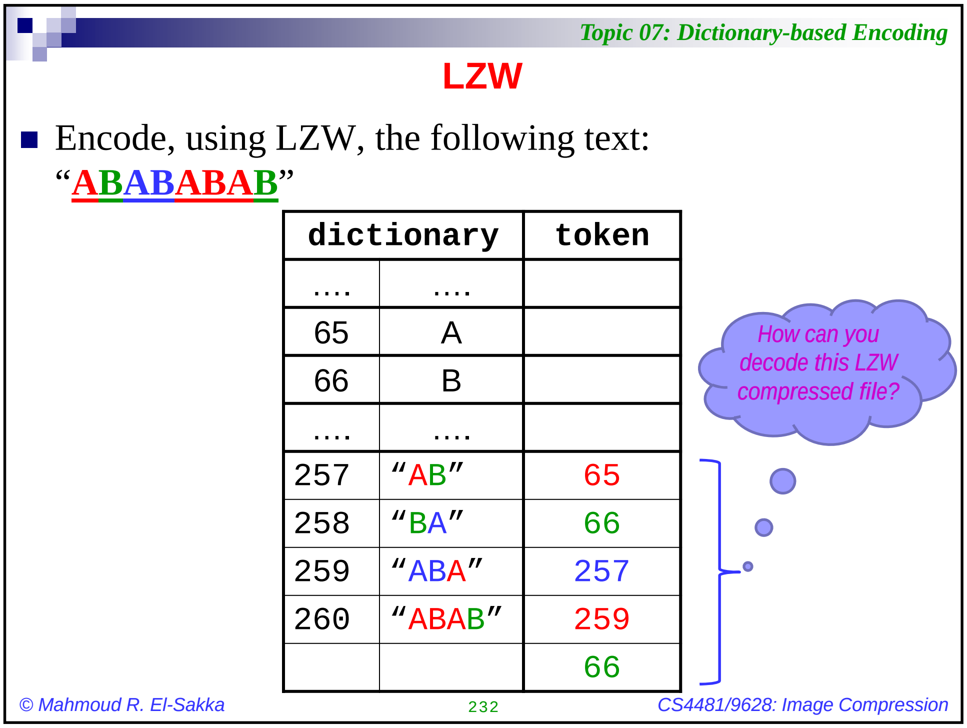
dictionary		token
….	….	
65	A	
66	B	
….	….	
257	"AB"	65
258	"BA"	66
259	"ABA"	257



LZW

- Encode, using LZW, the following text: “**ABABABA**B”

dictionary		token
….	….	
65	A	
66	B	
….	….	
257	“AB”	65
258	“BA”	66
259	“ABA”	257
260	“ABAB”	259



LZW

- Encode, using LZW, the following text: “**ABABABAB**”

dictionary		token
….	….	
65	A	
66	B	
….	….	
257	“AB”	65
258	“BA”	66
259	“ABA”	257
260	“ABAB”	259
		66

How can you decode this LZW compressed file?

LZW

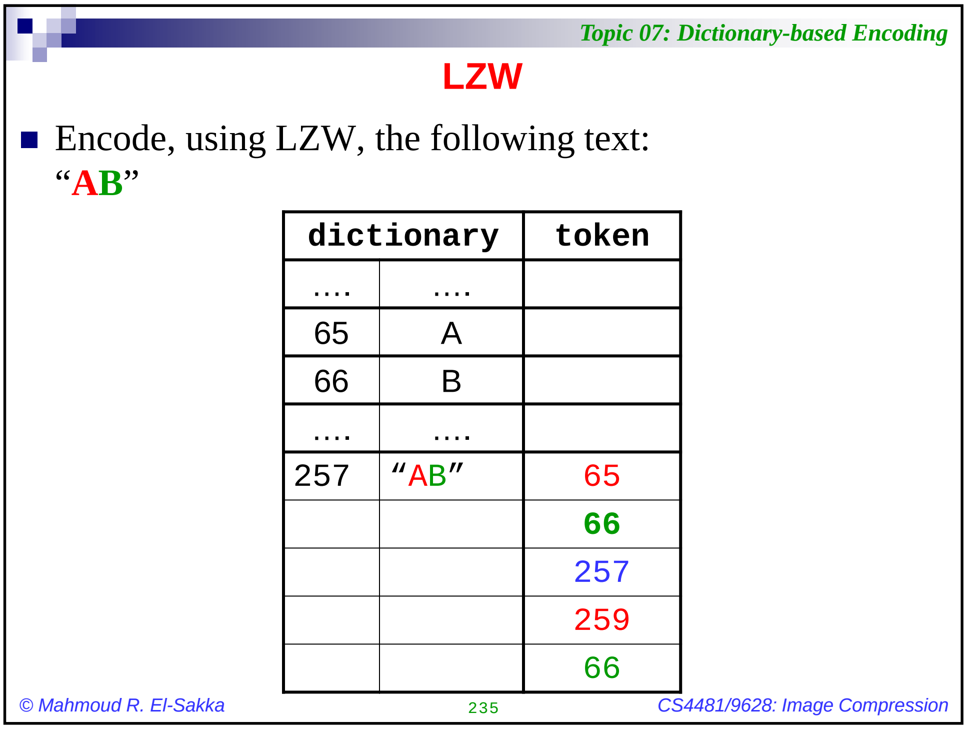
- Encode, using LZW, the following text:
  “”

dictionary		token
….	….	
65	A	
66	B	
….	….	
		65
		66
		257
		259
		66

LZW

- Encode, using LZW, the following text:
  “**A**”

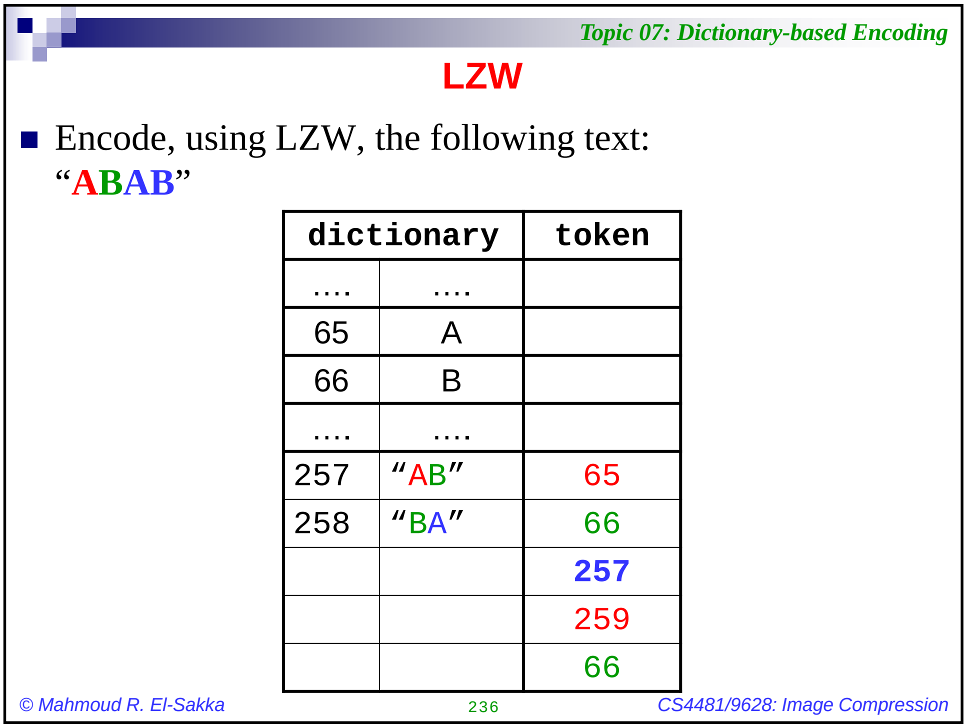
dictionary		token
….	….	
65	A	
66	B	
….	….	
		65
		66
		257
		259
		66



LZW

- Encode, using LZW, the following text: "AB"

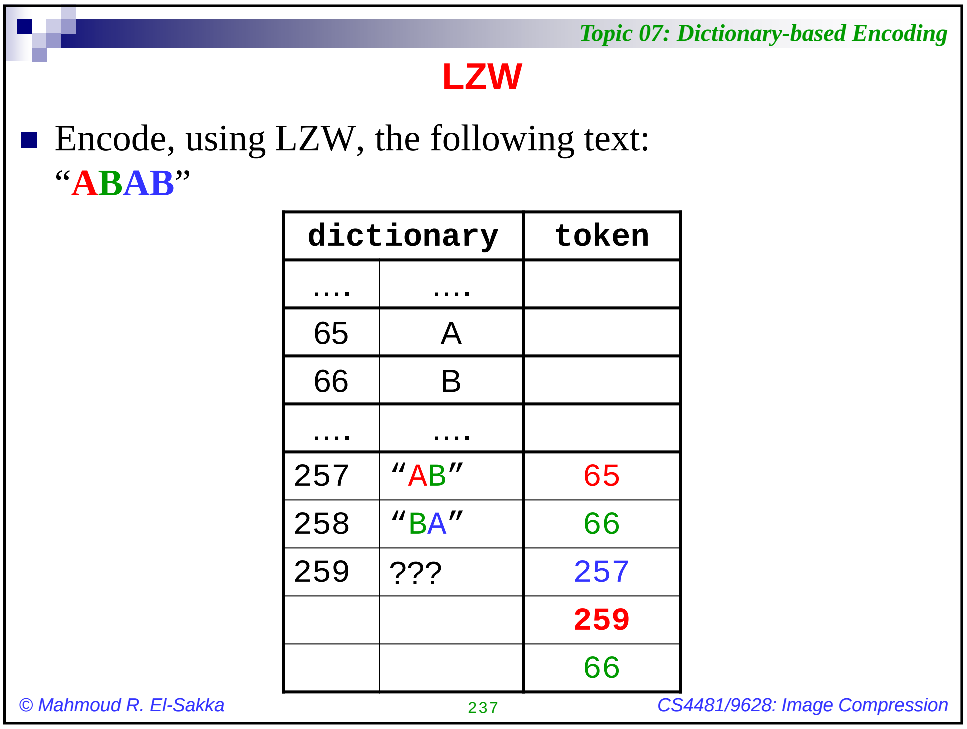
dictionary		token
….	….	
65	A	
66	B	
….	….	
257	"AB"	65
		66
		257
		259
		66



LZW

- Encode, using LZW, the following text: "ABAB"

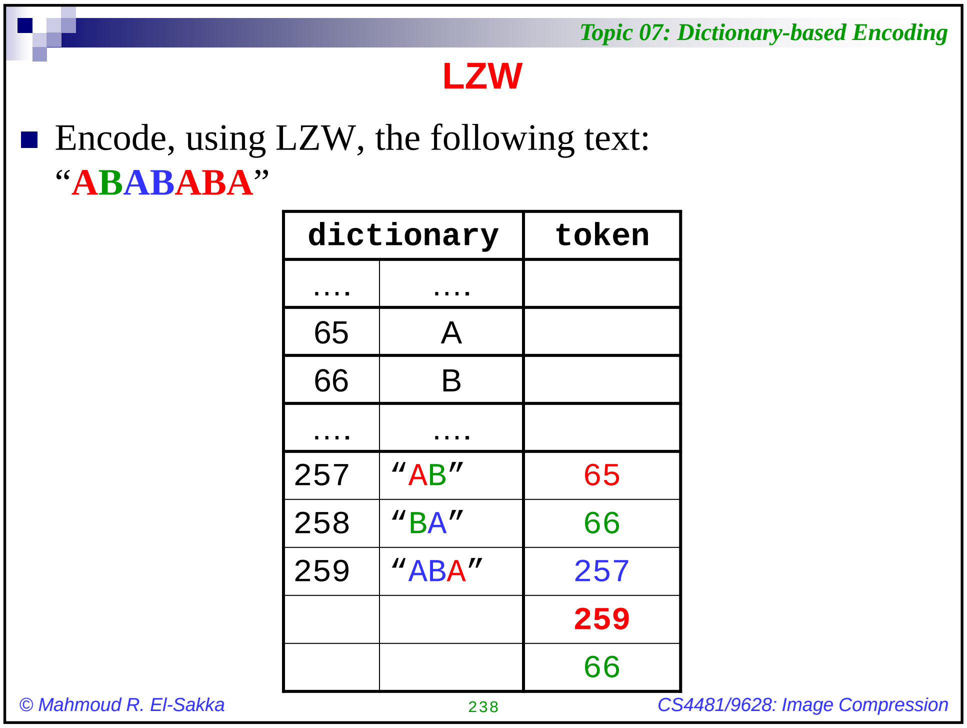
dictionary		token
….	….	
65	A	
66	B	
….	….	
257	"AB"	65
258	"BA"	66
		257
		259
		66



LZW

- Encode, using LZW, the following text: "**ABAB**"

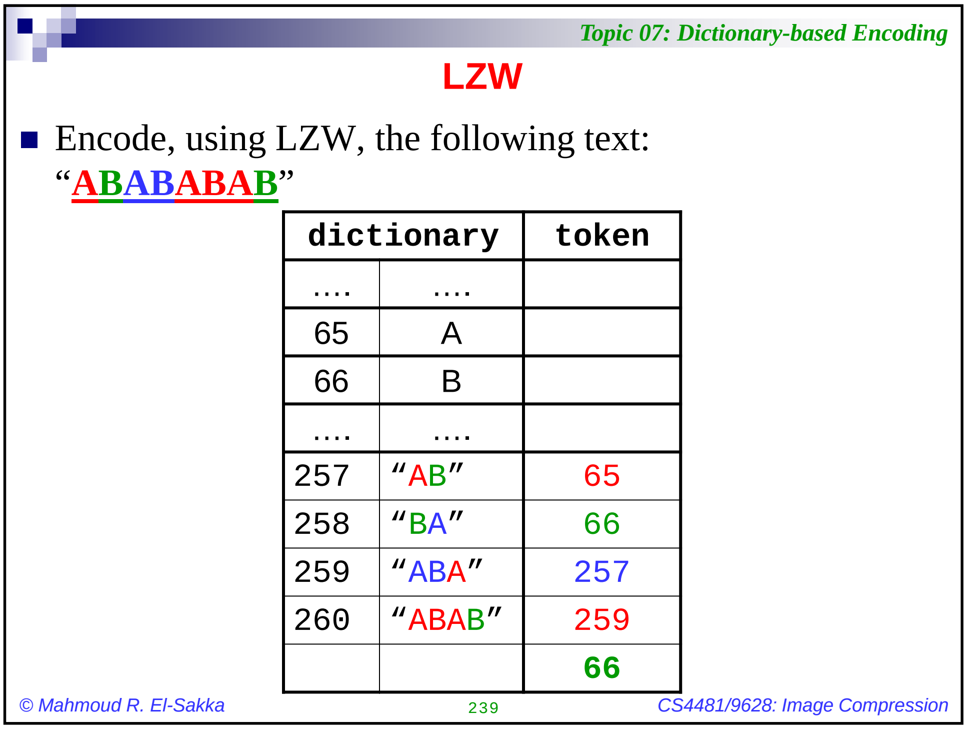
dictionary		token
….	….	
65	A	
66	B	
….	….	
257	"AB"	65
258	"BA"	66
259	???	257
		259
		66



LZW

- Encode, using LZW, the following text: "**ABABABA**"

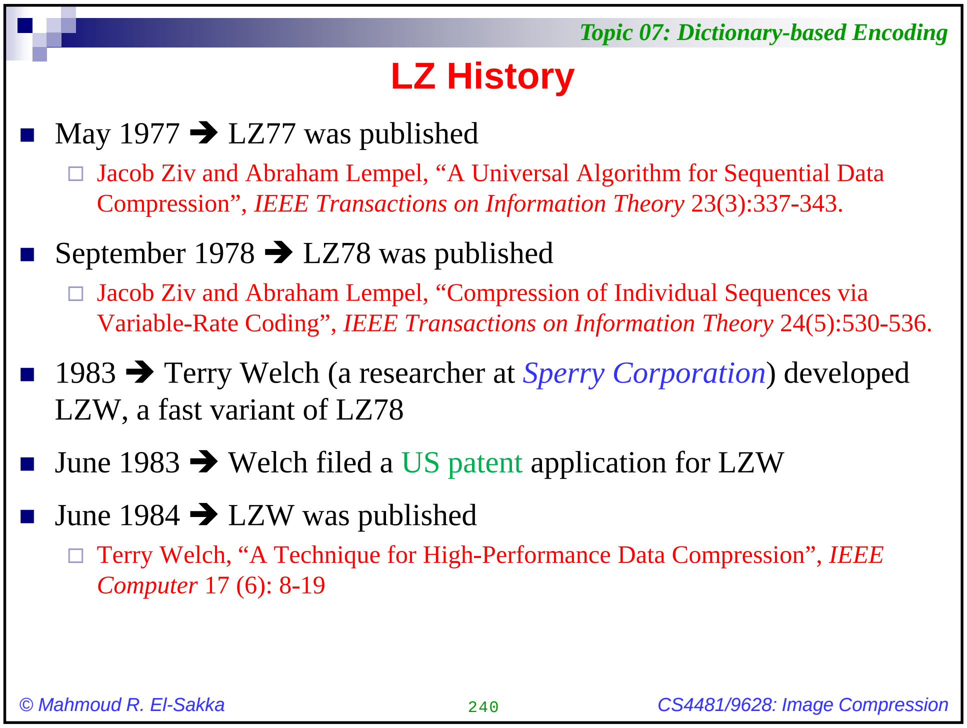
dictionary		token
….	….	
65	A	
66	B	
….	….	
257	"AB"	65
258	"BA"	66
259	"ABA"	257
		259
		66



LZW

- Encode, using LZW, the following text: “ABABABAB”

dictionary		token
….	….	
65	A	
66	B	
….	….	
257	“AB”	65
258	“BA”	66
259	“ABA”	257
260	“ABAB”	259
		66



LZ History

- May 1977 ➔ LZ77 was published
  - Jacob Ziv and Abraham Lempel, “A Universal Algorithm for Sequential Data Compression”, *IEEE Transactions on Information Theory* 23(3):337-343.
- September 1978 ➔ LZ78 was published
  - Jacob Ziv and Abraham Lempel, “Compression of Individual Sequences via Variable-Rate Coding”, *IEEE Transactions on Information Theory* 24(5):530-536.
- 1983 ➔ Terry Welch (a researcher at *Sperry Corporation*) developed LZW, a fast variant of LZ78
- June 1983 ➔ Welch filed a US patent application for LZW
- June 1984 ➔ LZW was published
  - Terry Welch, “A Technique for High-Performance Data Compression”, *IEEE Computer* 17 (6): 8-19

LZ History

- December 1985 ➔ LZW patent was granted (*US-4558302*) and assigned to *Sperry Corporation*
- 1986 ➔ *Sperry Corporation* merged with *Burroughs* and named ***Unisys***
  - □ *Unisys* became the holder of the LZW patent and entered into licensing agreements with hundreds of companies

GIF file format

- 1987 ➔ The *Graphic Interchange Format* (GIF) was developed by *Compuserve information Service* to encode ***graphical images***
  - □ GIF is a bitmap format that utilizes LZW *(i.e., lossless compression method)*
  - □ GIF uses palette of ***up to 256 different colors***
  - □ When dealing with images that have *less* that 256 different colors (e.g., *graphics or logos*)
    - In such cases ➔ *GIF becomes a lossless scheme*
  - □ When dealing with images that have *more* that 256 different colors (e.g., *natural scene images*)
    - GIF becomes incapable of representing full color
    - GIF quantizes the colors in an image to ***256 different colors,*** creating a palette of *quantized colors*
      - □ The colors in the palette are ***chosen from the 24-bit RGB color space*** and
      - □ mapped (i.e., indexed) to an ***8-bit index***
      - □ This color quantization is a *lossy process*,
    - In such cases ➔ *GIF becomes a lossy scheme*

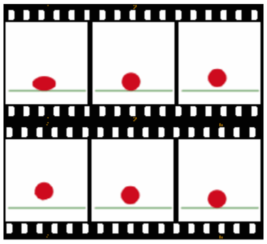
GIF file format

- □ The initial dictionary size is set to 512 entries
- □ The size of the dictionary is progressively doubled as it fills up until it becomes 4096 entries
- □ When the dictionary reaches the 4096 entries level, GIF behaves like a static dictionary

- 1989 ➔ Basic animation was added to the GIF89a (*the rapid display of a sequence of static images that minimally differ from each other*)
  - □ An animated GIF file consists of a number of images (frames)
  - □ These images are displayed in succession with short delay between them

GIF file format

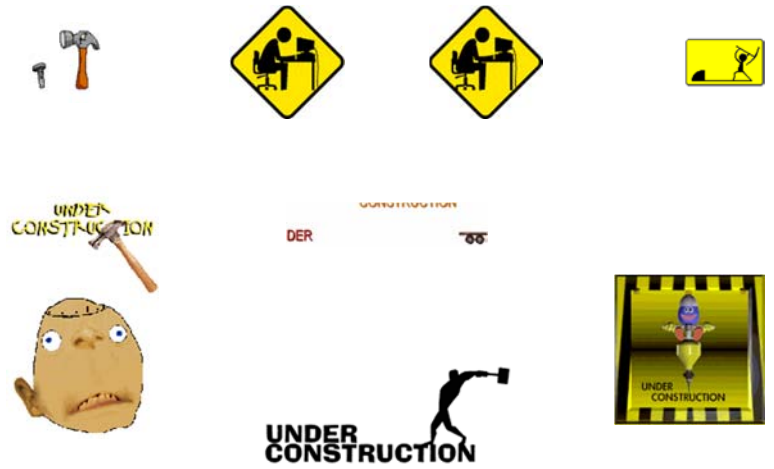
- Animation Example:
  - □ The following six 82×82 images are encoded inside an animated GIF file
  - □ This animation moves at 10 frames per second, i.e., the entire six images are looped each 0.6 second



- There are many free software tools than can generate such animated GIF images

GIF file format

- Animation Examples:



GIF file format

- Animation Example:

GIF file format

- Animation Example:

GIF file format

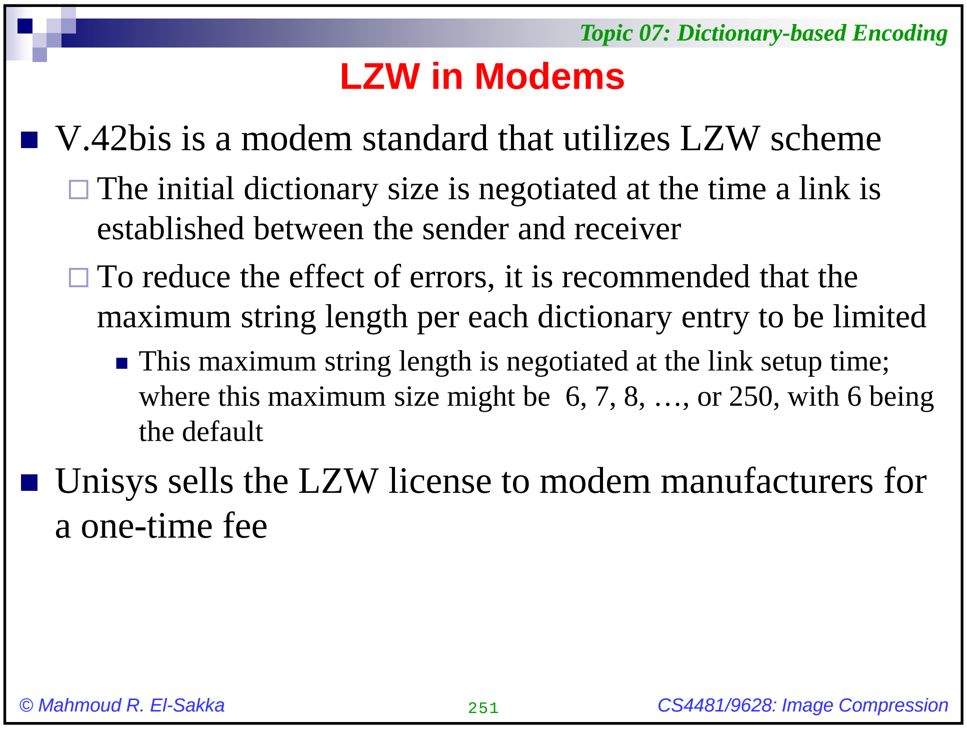
- At that time, CompuServe was not aware of the Unisys LZW patent
- 1992 ➔ *Unisys* (the LZW patent holder) became aware of the use of LZW in the GIF format
- January 1993 ➔ Unisys entered into licensing negotiations with CompuServe
- December 1994 ➔ *Compuserve* licensed the usage of LZW in GIF
- Unisys stated that
  - □ all major commercial on-line information services companies utilizing the LZW need to license the technology from Unisys at a reasonable fee
  - □ they would not require licensing, or fees to be paid, for non-commercial, non-profit GIF-based applications, including those for use on the on-line services

GIF file format

- Many software developers threatened to stop using the GIF format
- Due to these issues, the *Portable Network Graphic* (PNG) *Development Group* has been formed and within a year and half, the final version of the PNG format has been released in October 1, 1996

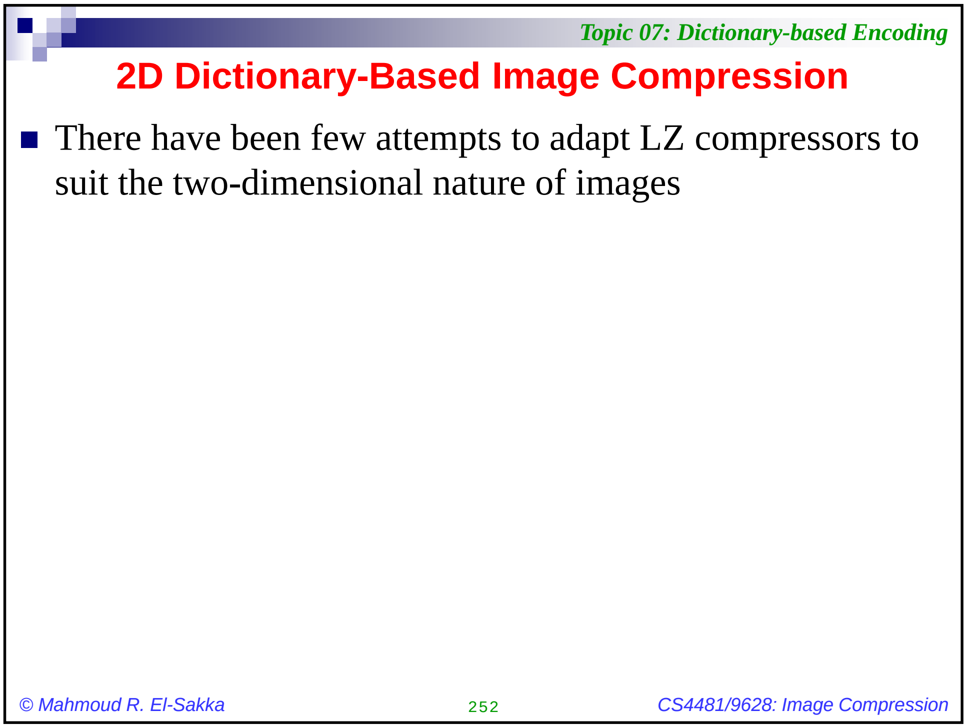
GIF file format

- August 1999 ➔ Unisys announced the option for owners of certain non-commercial and private websites to obtain licenses on payment of a one-time license fee of $5000
  - Unisys was subjected to thousands of online attacks and abusive emails from users believing that they were going to be charged $5000 or sued for using GIFs on their websites
- Unisys was completely unable to generate any good publicity and continued to be condemned by individuals and organizations
  - The *League for Programming Freedom* started the "***Burn All GIFs***" campaign
- June 2003 ➔ The United States LZW patent expired
- July 2004 ➔ The United Kingdom, France, Germany Italy, Japan, Canada LZW patents expired as well
  - Consequently, the GIF format may now be freely used



LZW in Modems

- V.42bis is a modem standard that utilizes LZW scheme
  - □ The initial dictionary size is negotiated at the time a link is established between the sender and receiver
  - □ To reduce the effect of errors, it is recommended that the maximum string length per each dictionary entry to be limited
    - This maximum string length is negotiated at the link setup time; where this maximum size might be 6, 7, 8, …, or 250, with 6 being the default
- Unisys sells the LZW license to modem manufacturers for a one-time fee

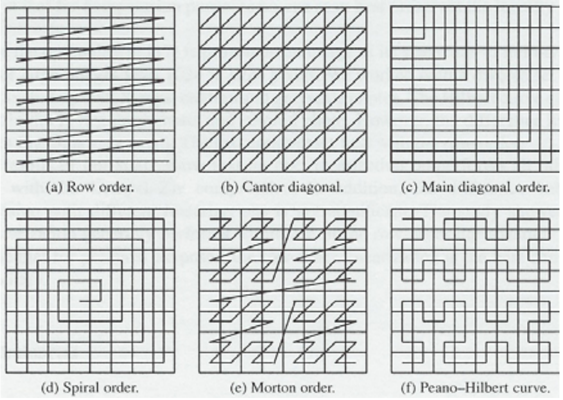


2D Dictionary-Based Image Compression

- There have been few attempts to adapt LZ compressors to suit the two-dimensional nature of images

2D Dictionary-Based Image Compression

- The first attempt was to find a way to linearize the data and then use a one-dimensional compressor on the data



(a) Row order. (b) Cantor diagonal. (c) Main diagonal order.

(d) Spiral order. (e) Morton order. (f) Peano–Hilbert curve.

- Tests show that no one linearization is best for all images

2D Dictionary-Based Image Compression

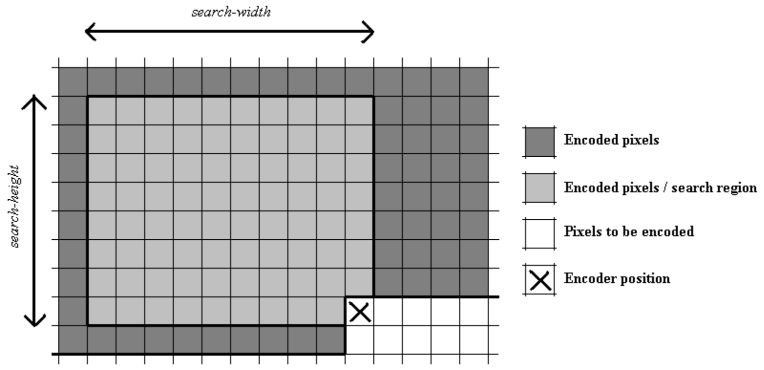
- Few years ago, there was no compression scheme exists which considers the two-dimensional nature of images, other than binary images
- In 2005, a grayscale two-dimensional Lempel-Ziv image compression scheme (denoted GS-2D-LZ) was introduced by N. Brittain & M. El-Sakka
  - □ Nathanael J. Brittain and Mahmoud R. El-Sakka, “Grayscale True Two-Dimensional Dictionary-based Image Compression”, *Journal of Visual Communication & Image Representation*, 18(1), pp. 35--44, February 2007.

GS-2D-LZ Overview

- An image is encoded in a raster scan order processing, one block of pixels at each step
- For each block of pixels an approximate match is searched for in previously encoded data
- Each block is replaced with the codeword for the approximate match

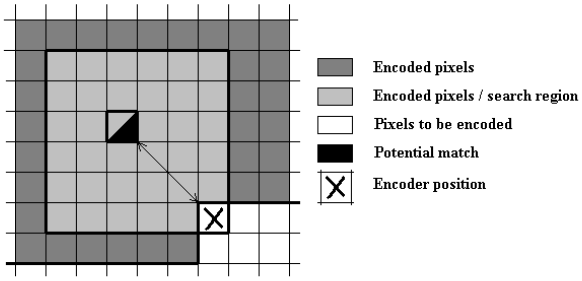
Finding and Encoding Matches

- The search area is a rectangular region above and to the left of the encoder position
- Each pixel in the search region represents the root of a potential match



Finding and Encoding Matches

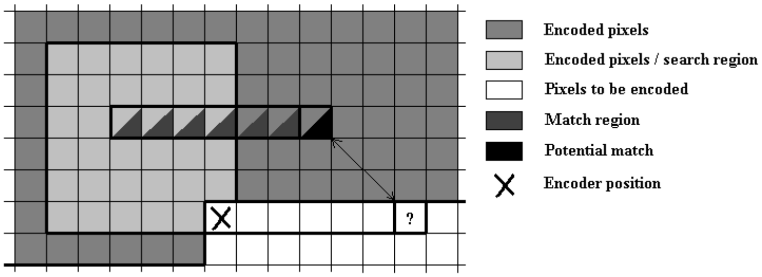
- To be considered as a possible match, the difference of a particular root in the search region and the pixel at the encoder position cannot exceed the value of *threshold* (the maximum allowable error between pixels)



Absolute Value (◪ - ☒) < *Threshold* ?

Finding and Encoding Matches

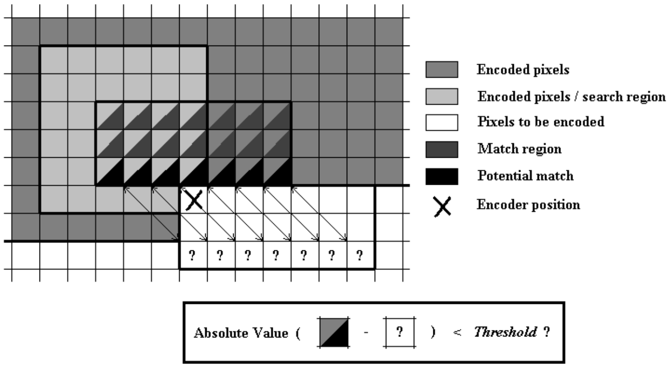
- Matches are then extended to the right as far as possible
- To extend one pixel to the right, the difference between corresponding pixels must be less than the value of *threshold*



Absolute Value (◪ - ?) < *Threshold* ?

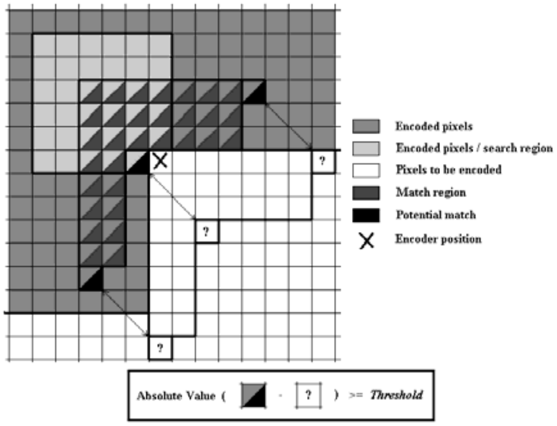
Finding and Encoding Matches

- The match is then extended as far down as possible
- For each attempt to extend the height of the match by one, a row of corresponding pixels will be evaluated for a potential match



Matching Criteria

- More than one match may exist at the same root in the search region
- The match with the largest number of new pixels is taken



Finding and Encoding Matches

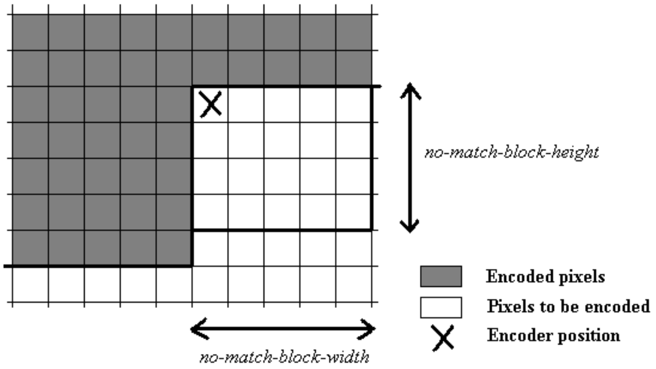
- Matches are evaluated by their size and by mean square error (the average squared difference between pixels)
- If the match is good enough, its dimensions and offset are recorded in tables
- Residual data is also recorded in a table

Matching Criteria

- A match must be large enough (more new pixels being encoded than the value of a variable called *minimum-match-size*)
- In addition, if the MSE of the new pixels of the potential match is less than the variable *max-MSE*, the match is considered sufficient

No Match

- When no sufficient match is found within the search region, a small block of pixels rooted at the encoder position is encoded using a predictive encoding
- Prediction errors are recorded in a table



Data Structures

- In addition to the match offset, match dimensions, match residual, and prediction error tables, a match flag table is used to distinguish between matches and mismatches
- After the entire image has been processed, each of the tables are encoded using a statistical encoder

Data Structures

- A benchmark of statistical encoders was made to determine the best one to use with GS-2D-LZ
- PAQ6, a context-based arithmetic encoder, was chosen
- Although PAQ6 was the slowest encoder of those tested, it produced the best compression

Training GS-2D-LZ

- Variables in GS-2D-LZ determine the search region and the characteristics of each match
- The optimal setting for each variable was determined by adjusting its value from a minimum to a maximum value while fixing each other variable
- The training set used to optimize the variables is composed of 6 images from each of 4 classes: geographic, graphic, natural scene, and medical
- None of the training images were used in the final tests

Sample Images Used in Training GS-2D-LZ

Experimental Set-up

- GS-2D-LZ was tested on a set of 110 gray scale images
  - 24 geographic images
  - 24 graphic images
  - 24 natural scene images
  - 24 medical images
  - 16 standard test images
- Two group of experiments were performed
  - Tests versus Dictionary-Based Schemes
  - Tests versus State-of-the-Art Schemes

Tests versus Dictionary-Based Schemes

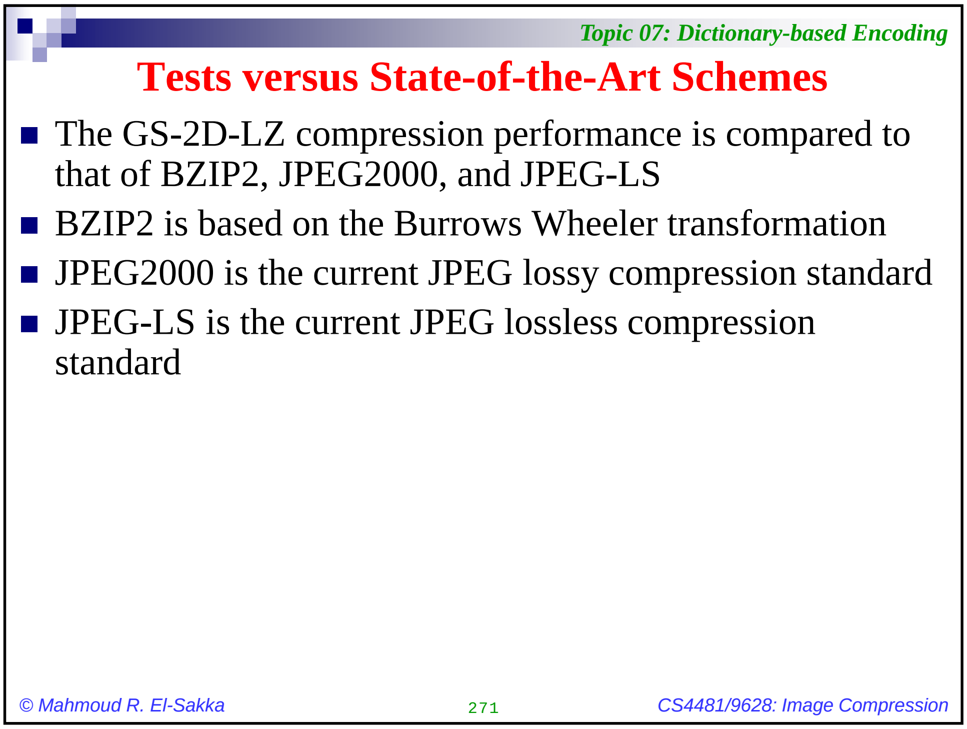
- The GS-2D-LZ compression performance is compared to that of PNG, GIF, and Unix Compress
- PNG is based on LZ77, whereas Unix Compress and GIF are based on a variation of LZ78

Tests versus Dictionary-Based Schemes

Image Class	GS-2D-LZ	PNG	GIF	Unix Compress
geography	5.17	5.40	7.37	6.46
graphic	1.61	1.77	2.73	2.48
natural	4.50	4.73	6.83	6.21
medical	3.25	3.42	4.96	4.58
standard	4.54	4.80	6.99	6.65

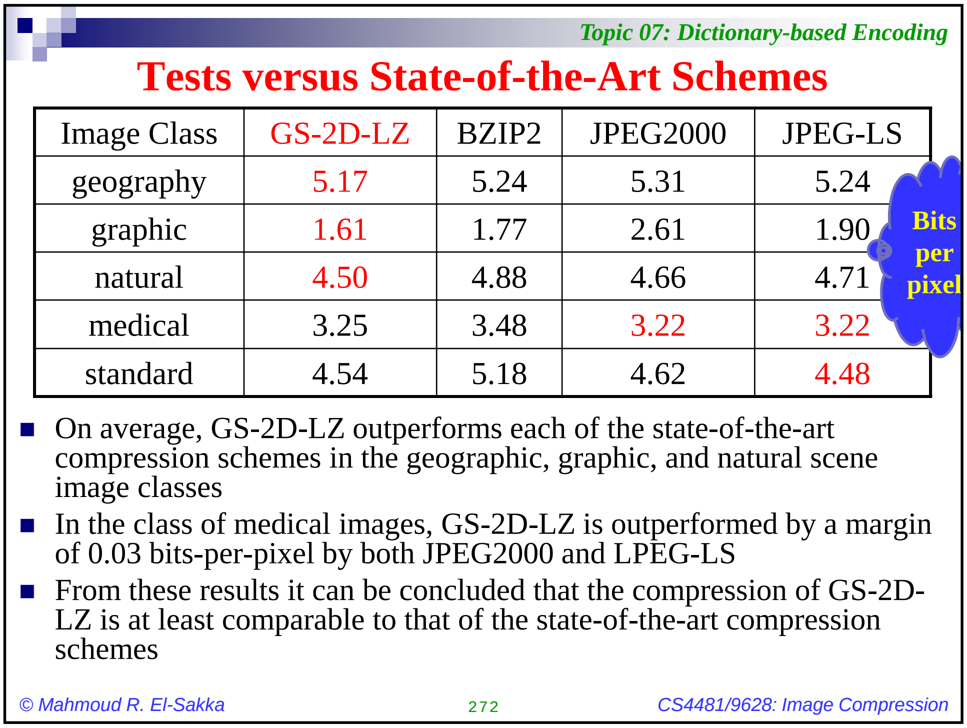
Bits per pixel

- On average, GS-2D-LZ outperforms each of the other dictionary-based compression schemes in each of the image classes and on the set of standard test images
- From these results, we can conclude that GS-2D-LZ surpasses the compression performance of any other dictionary-based scheme



Tests versus State-of-the-Art Schemes

- The GS-2D-LZ compression performance is compared to that of BZIP2, JPEG2000, and JPEG-LS
- BZIP2 is based on the Burrows Wheeler transformation
- JPEG2000 is the current JPEG lossy compression standard
- JPEG-LS is the current JPEG lossless compression standard

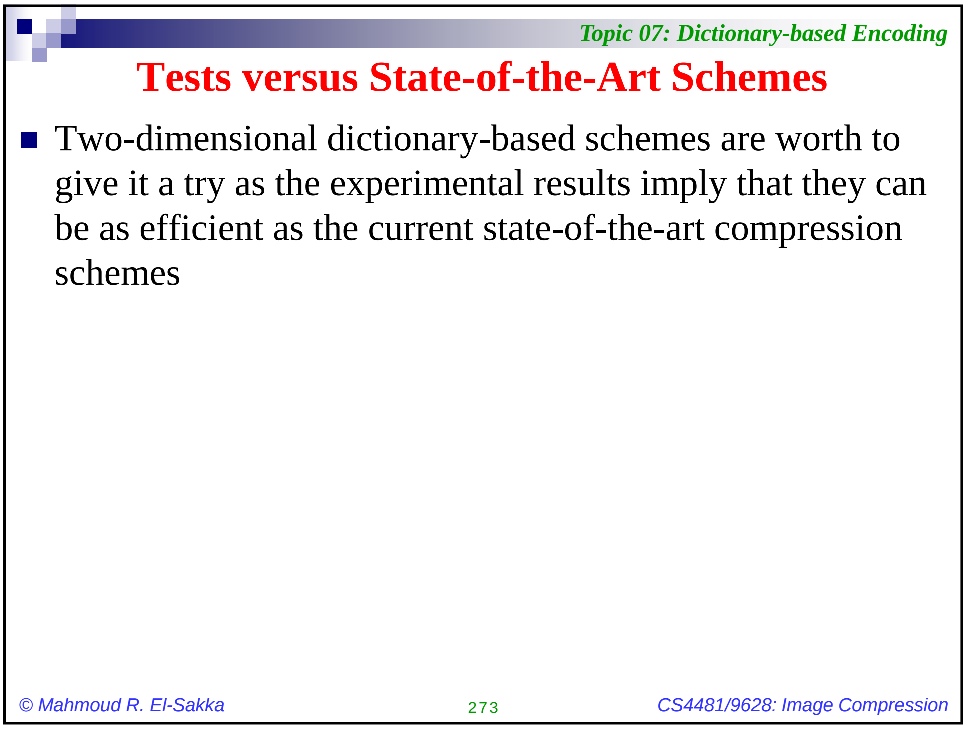


Tests versus State-of-the-Art Schemes

Image Class	GS-2D-LZ	BZIP2	JPEG2000	JPEG-LS
geography	5.17	5.24	5.31	5.24
graphic	1.61	1.77	2.61	1.90
natural	4.50	4.88	4.66	4.71
medical	3.25	3.48	3.22	3.22
standard	4.54	5.18	4.62	4.48

Bits per pixel

- On average, GS-2D-LZ outperforms each of the state-of-the-art compression schemes in the geographic, graphic, and natural scene image classes
- In the class of medical images, GS-2D-LZ is outperformed by a margin of 0.03 bits-per-pixel by both JPEG2000 and LPEG-LS
- From these results it can be concluded that the compression of GS-2D-LZ is at least comparable to that of the state-of-the-art compression schemes



Tests versus State-of-the-Art Schemes

- Two-dimensional dictionary-based schemes are worth to give it a try as the experimental results imply that they can be as efficient as the current state-of-the-art compression schemes