Multimedia
Lecture 2

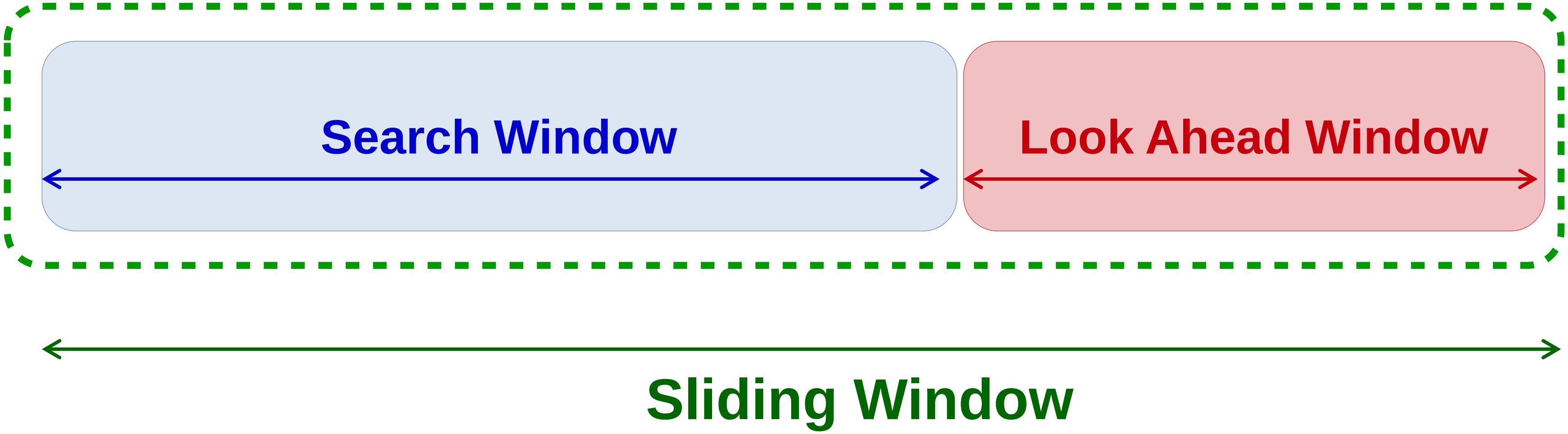
Dr. Mona M.Soliman

Faculty of Computers and Artificial Intelligence

Cairo University

Fall 2022

Recall: Lempel Ziv 77 Algorithm



Search Window

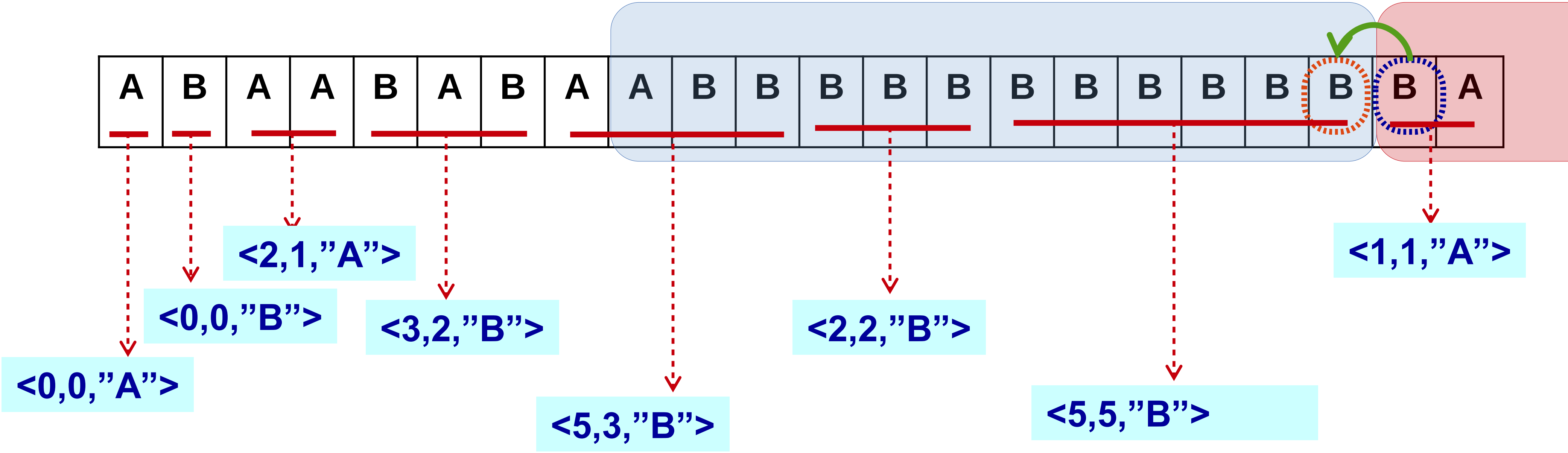
Look Ahead Window

Sliding Window

A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	B	A

TAG → <Position , Length , Next Symbol >

Recall: LZ 77 (Compression)



A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	B	A

<0,0,"A">

<0,0,"B">

<2,1,"A">

<3,2,"B">

<5,3,"B">

<2,2,"B">

<5,5,"B">

<1,1,"A">

Tag = < Position, Length ,Next Symbol Code>

Advantages and Disadvantage of LZ77

Advantages of LZ77

- Probabilities of symbols is not required to be known a priori. (suitable for Real time Compression).
- That is, the longer the size of the sliding window, the better the performance of data compression
- No coding table Required for Decompression.

Disadvantage of LZ77

A straightforward implementation would require up to [Look Ahead Buffer Size] * [Search Window Size] Symbol comparisons per Tag produced. Complexity of comparison is very large

LZ 78 (Compression)

A	B	A	A	B	A	B	A	A	B	A	B	B	B	B	B	B	B	B	B	B	A

Compress the Following Text (22 Characters)

Custom Dictionary (first Word is reserved as Empty)

0	------
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

TAG → **<Index in dictionary , Next Symbol >**

LZ 78 (Compression)

A	B	A	A	B	A	B	A	A	B	A	B	B	B	B	B	B	B	B	B	B	A

< 0 , “A” >

0	------
1	A
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

“A” is not in the dictionary

Save “A” as < 0,”A”>

Add Symbol=”A” to Dictionary

LZ 78 (Compression)

A	B	A	A	B	A	B	A	A	B	A	B	B	B	B	B	B	B	B	B	B	A

< 0 , “A” >

< 0 , “B” >

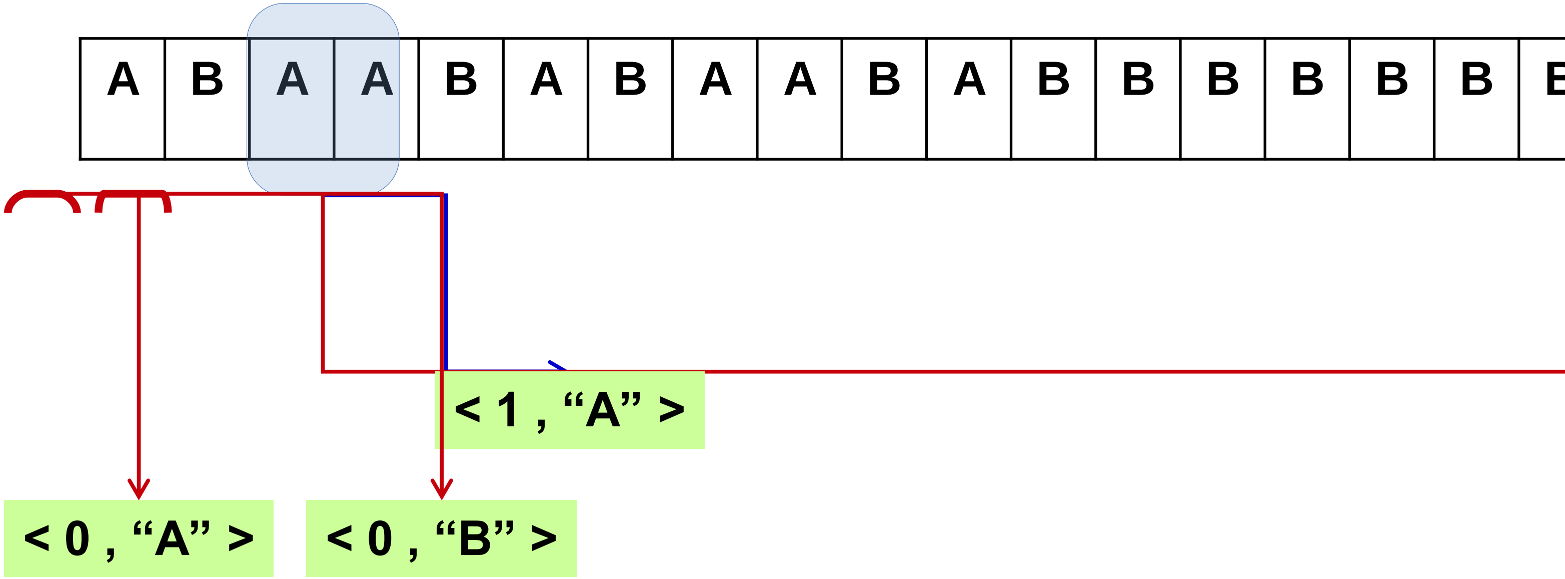
0	------
1	A
2	B
3	
4	
5	
6	
7	
8	
9	
10	
11	

“B” is not in the dictionary

Save “B” as < 0,”B”>

Add Symbol=”B” to Dictionary

LZ 78 (Compression)



A	B	A	A	B	A	B	A	A	B	A	B	B	B	B	B	B	B	B	B	B	A

< 1 , “A” >

< 0 , “A” >

< 0 , “B” >

0	------
1	A
2	B
3	AA
4	
5	
6	
7	
8	
9	
10	
11	

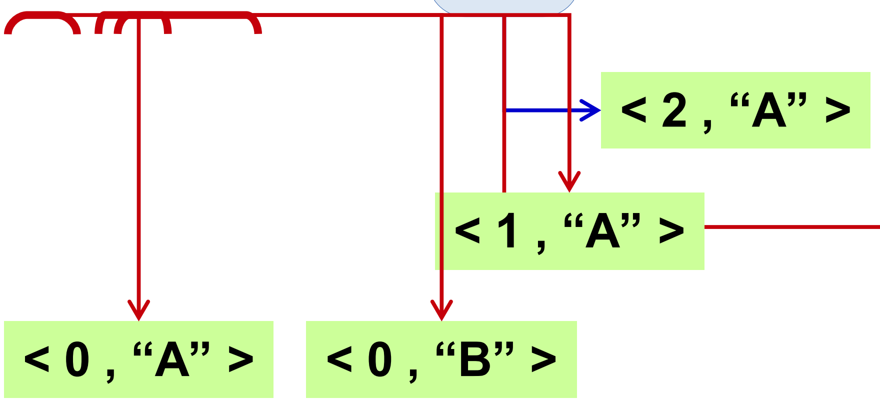
“A” is in the dictionary BUT “AA” is NOT

Save “AA” as < 1,”A”>

Add Symbols=”AA” to Dictionary

LZ 78 (Compression)

A	B	A	A	B	A	B	A	A	B	A	B	B	B	B	B	B	B	B	B	B	A



< 2 , “A” >

< 1 , “A” >

< 0 , “A” >

< 0 , “B” >

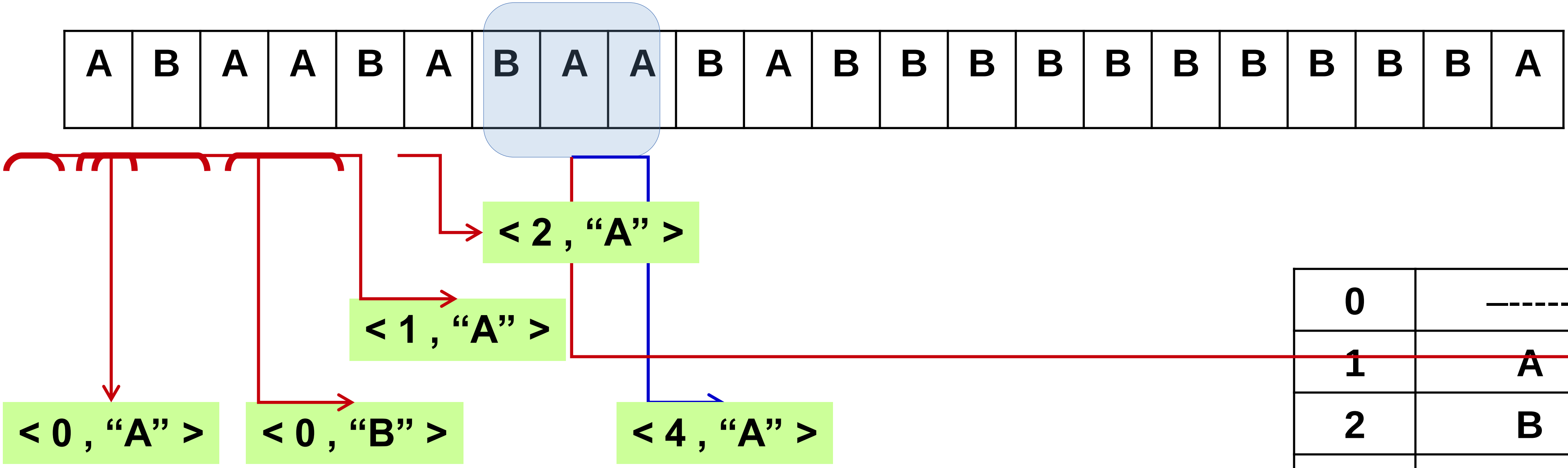
0	------
1	A
2	B
3	AA
4	BA
5	
6	
7	
8	
9	
10	
11	

“B” is in the dictionary BUT “BA” is NOT

Save “BA” as < 2,”A”>

Add Symbols=”BA” to Dictionary

LZ 78 (Compression)



A B A A B A B A A B A B B B B B B B B B B A

< 2 , “A” >

< 1 , “A” >

< 0 , “A” >

< 0 , “B” >

< 4 , “A” >

0	------
1	A
2	B
3	AA
4	BA
5	BAA
6	
7	
8	
9	
10	
11	

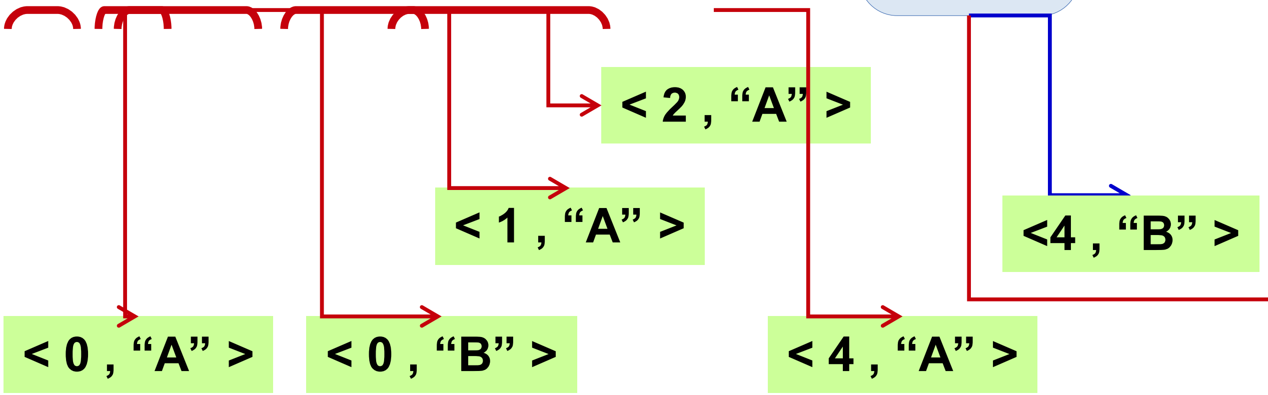
“BA” is in the dictionary BUT “BAA” is NOT

Save “BAA” as < 4,”A”>

Add Symbols=”BAA” to Dictionary

LZ 78 (Compression)

A	B	A	A	B	A	B	A	A	B	A	B	B	B	B	B	B	B	B	B	B	A



< 0 , “A” >

< 0 , “B” >

< 1 , “A” >

< 2 , “A” >

< 4 , “A” >

<4 , “B” >

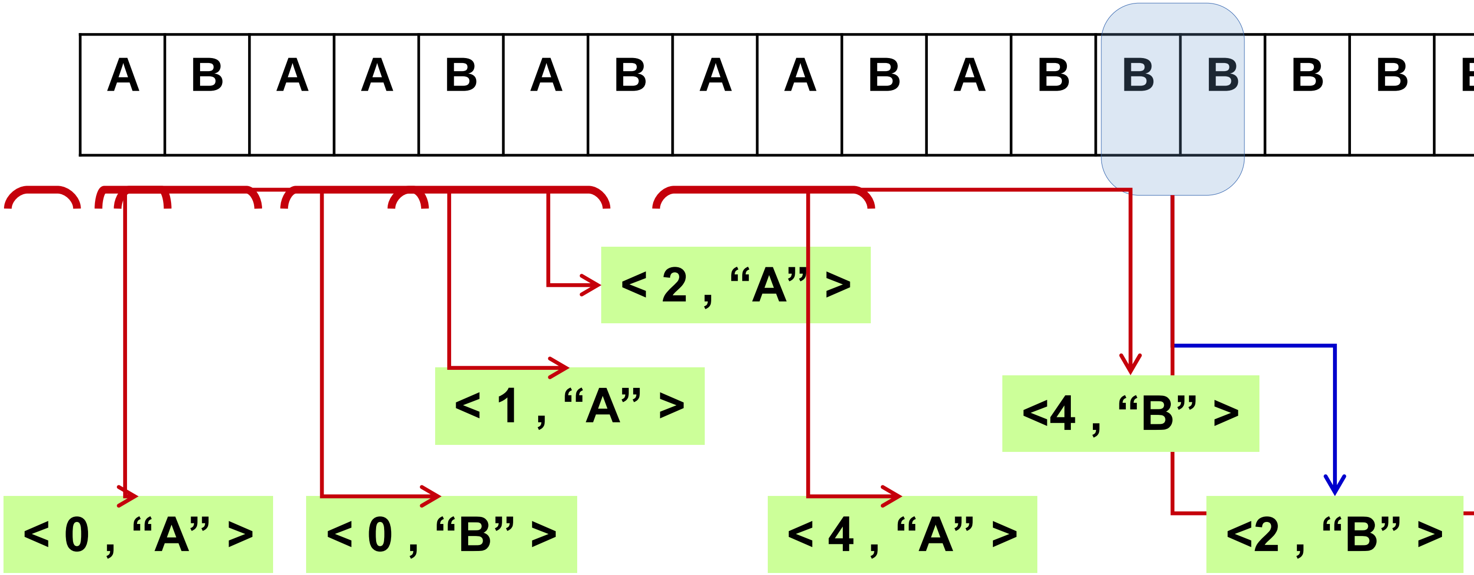
0	------
1	A
2	B
3	AA
4	BA
5	BAA
6	BAB
7	
8	
9	
10	
11	

“BA” is in the dictionary BUT “BAB” is NOT

Save “BAA” as < 4,”B”>

Add Symbols=”BAB” to Dictionary

LZ 78 (Compression)



A	B	A	A	B	A	B	A	A	B	A	B	B	B	B	B	B	B	B	B	B	A

< 2 , “A” >

< 1 , “A” >

<4 , “B” >

< 0 , “A” >

< 0 , “B” >

< 4 , “A” >

<2 , “B” >

0	------
1	A
2	B
3	AA
4	BA
5	BAA
6	BAB
7	BB
8	
9	
10	
11	

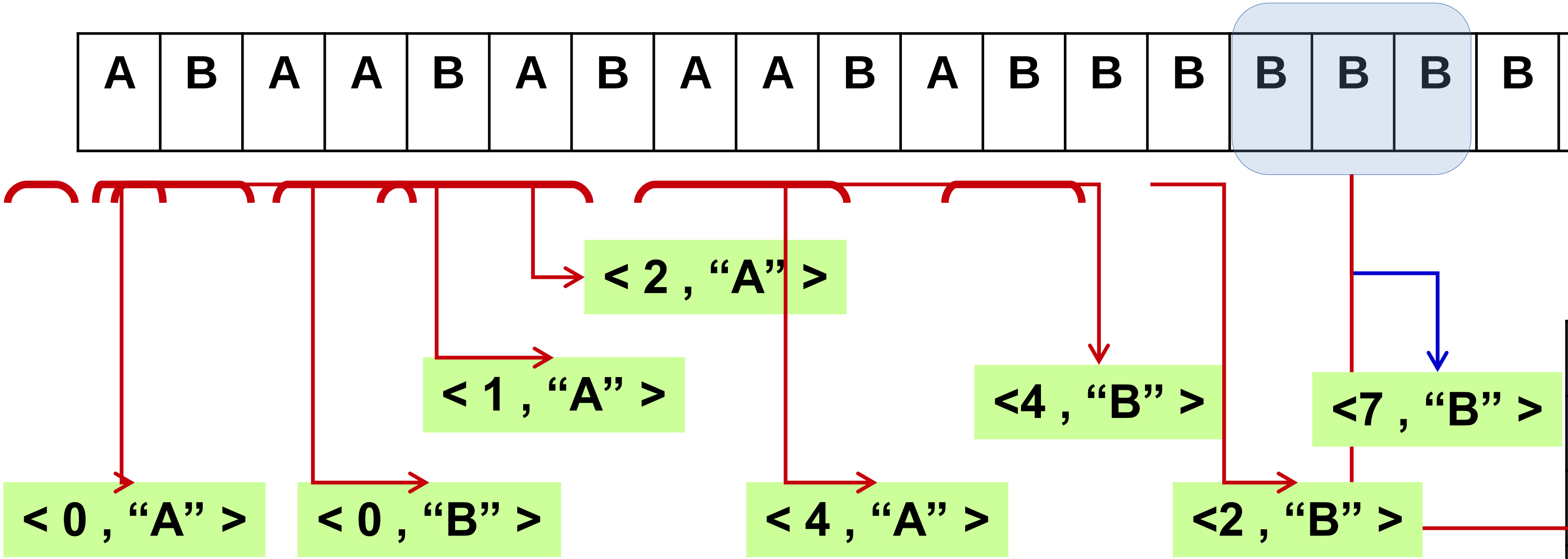
“B” is in the dictionary BUT “BB” is NOT

Save “BB” as < 2,”B”>

Add Symbols=”BB” to Dictionary

LZ 78 (Compression)

A	B	A	A	B	A	B	A	A	B	A	B	B	B	B	B	B	B	B	B	B	A



< 2 , “A” >

< 1 , “A” >

<4 , “B” >

<7 , “B” >

< 0 , “A” >

< 0 , “B” >

< 4 , “A” >

<2 , “B” >

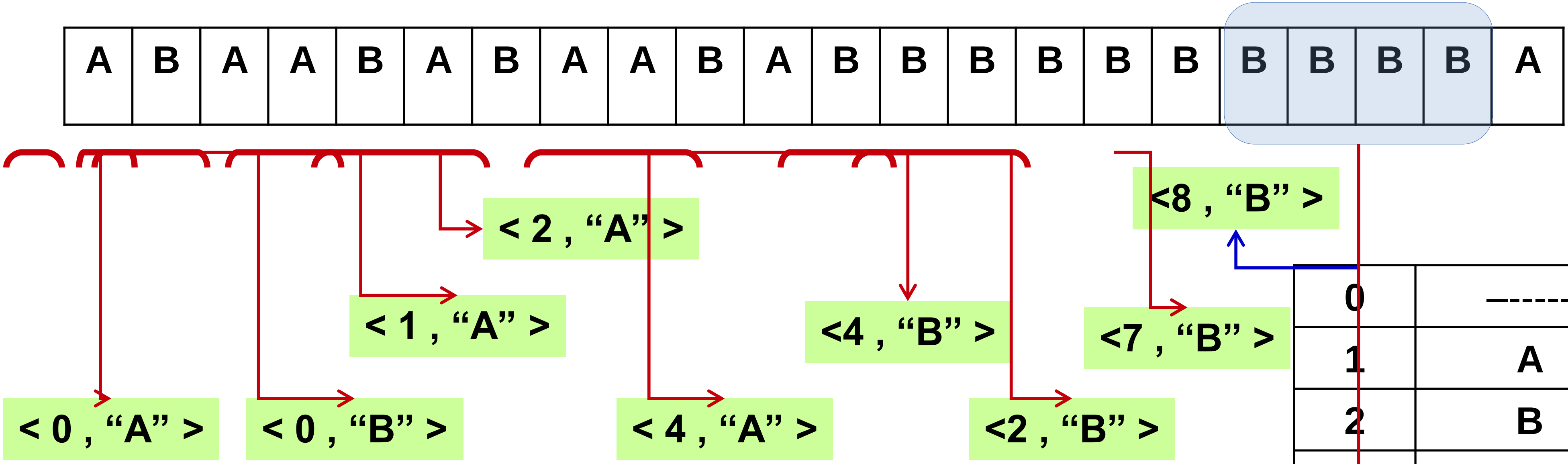
0	------
1	A
2	B
3	AA
4	BA
5	BAA
6	BAB
7	BB
8	BBB
9	
10	
11	

“BB” is in the dictionary BUT “BBB” is NOT

Save “BBB” as < 7,”B”>

Add Symbols=”BBB” to Dictionary

LZ 78 (Compression)



A	B	A	A	B	A	B	A	A	B	A	B	B	B	B	B	B	B	B	B	B	A

< 0 , “A” >

< 0 , “B” >

< 1 , “A” >

< 2 , “A” >

< 4 , “A” >

<4 , “B” >

<2 , “B” >

<7 , “B” >

<8 , “B” >

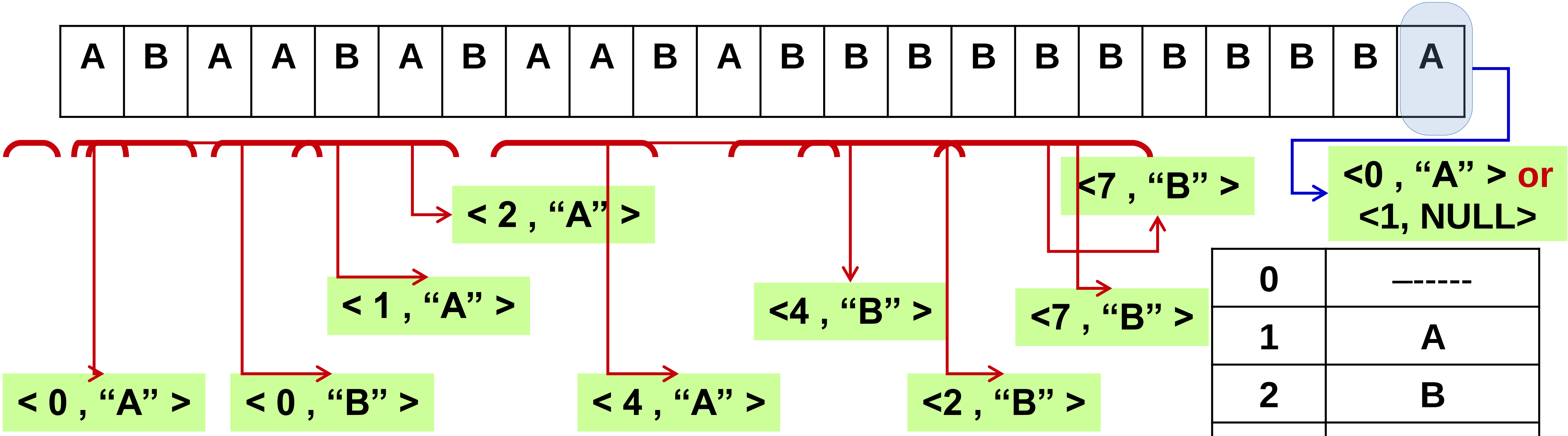
0	------
1	A
2	B
3	AA
4	BA
5	BAA
6	BAB
7	BB
8	BBB
9	BBBB
10	
11	

“BBB” is in the dictionary BUT “BBBB” is NOT

Save “BBBB” as < 8,”B”>

Add Symbols=”BBBB” to Dictionary

LZ 78 (Compression)



A	B	A	A	B	A	B	A	A	B	A	B	B	B	B	B	B	B	B	B	B	A

< 0 , “A” >

< 0 , “B” >

< 1 , “A” >

< 2 , “A” >

< 4 , “A” >

<4 , “B” >

<2 , “B” >

<7 , “B” >

<7 , “B” >

<0 , “A” > or <1, NULL>

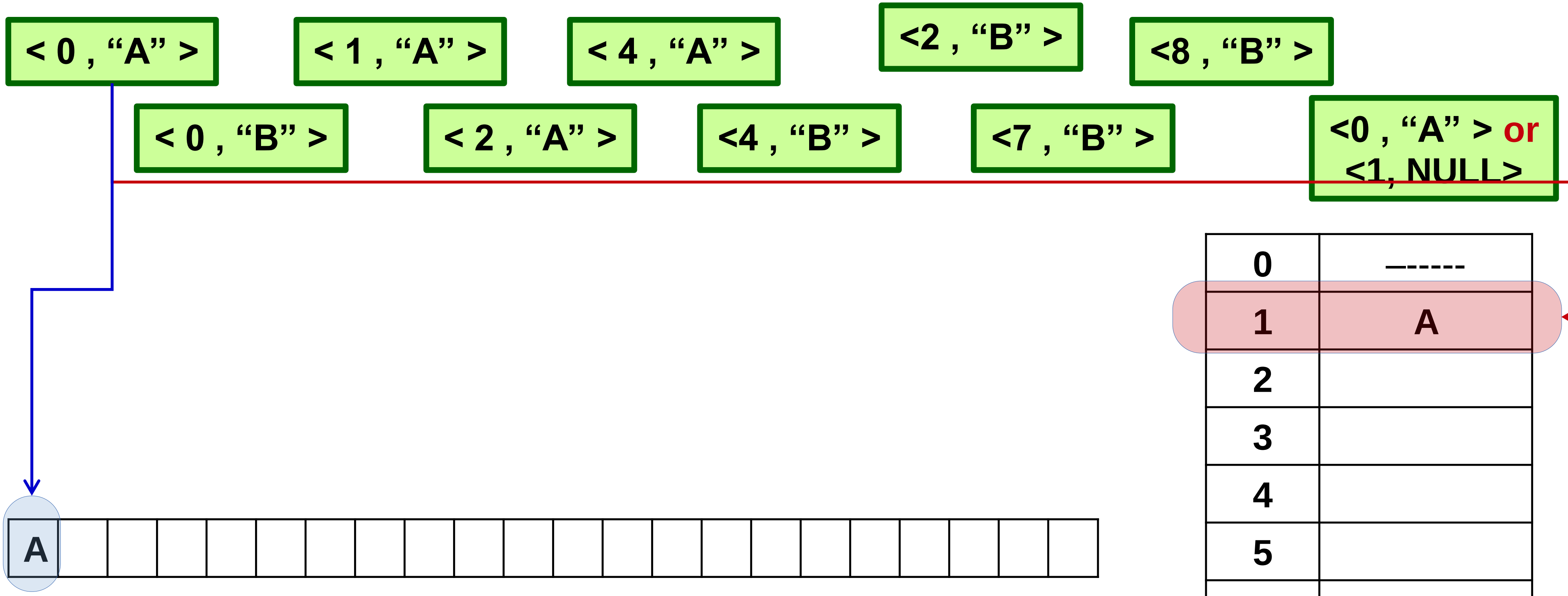
0	------
1	A
2	B
3	AA
4	BA
5	BAA
6	BAB
7	BB
8	BBB
9	BBBB
10	
11	

“A” is in the dictionary

Save “A” as <1 ,NULL> or < 0,”A”>

Add NOTHING to Dictionary

LZ 78 (De-Compression)

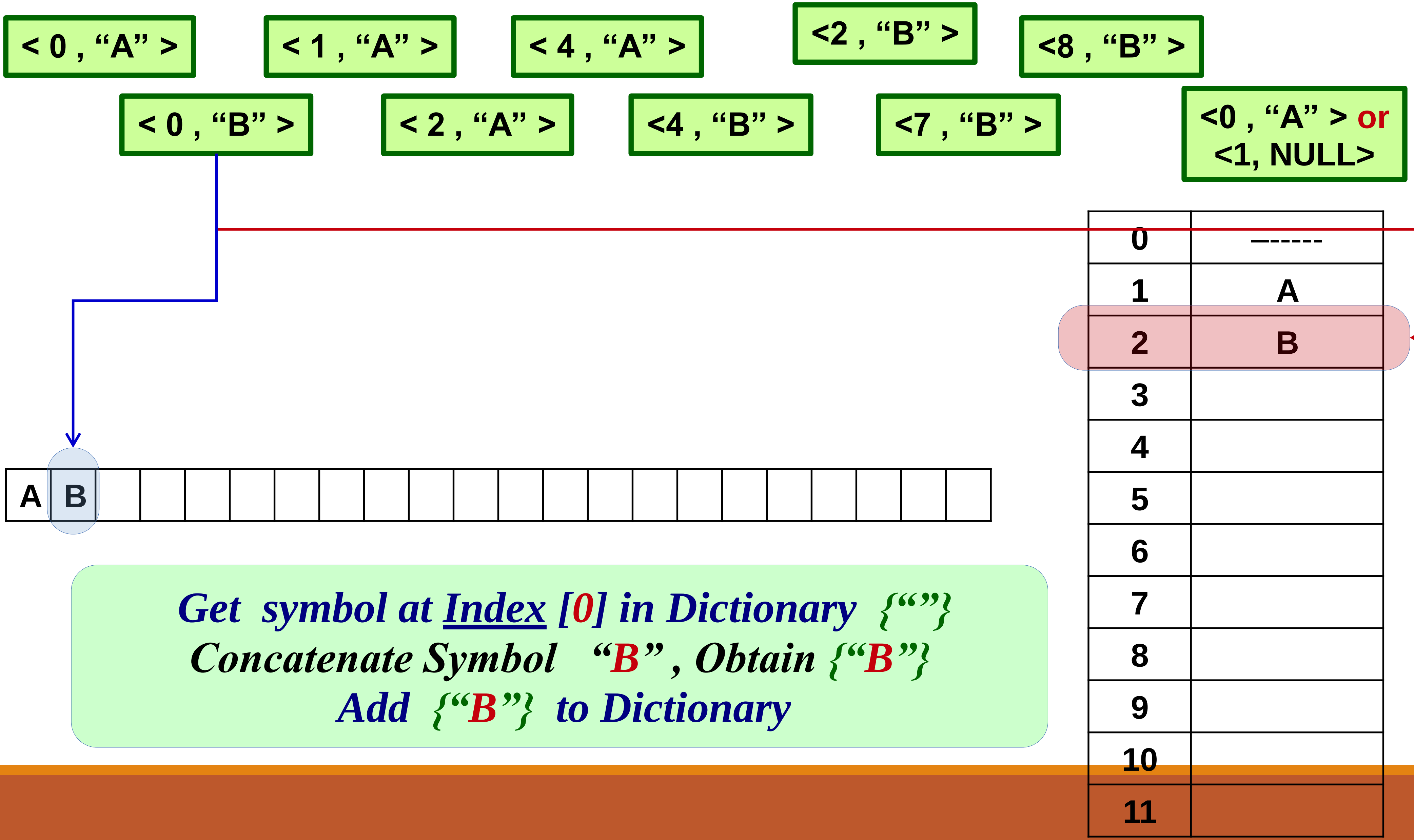


< 0 , “A” >
< 0 , “B” >
< 1 , “A” >
< 2 , “A” >
< 4 , “A” >
<4 , “B” >
<2 , “B” >
<7 , “B” >
<8 , “B” >
<0 , “A” > or <1, NULL>

0	–-----
1	A
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

Get symbol at ***Index*** *[0] in Dictionary {“”}*
Concatenate Symbol “A” , Obtain {“A”}
Add {“A”} to Dictionary

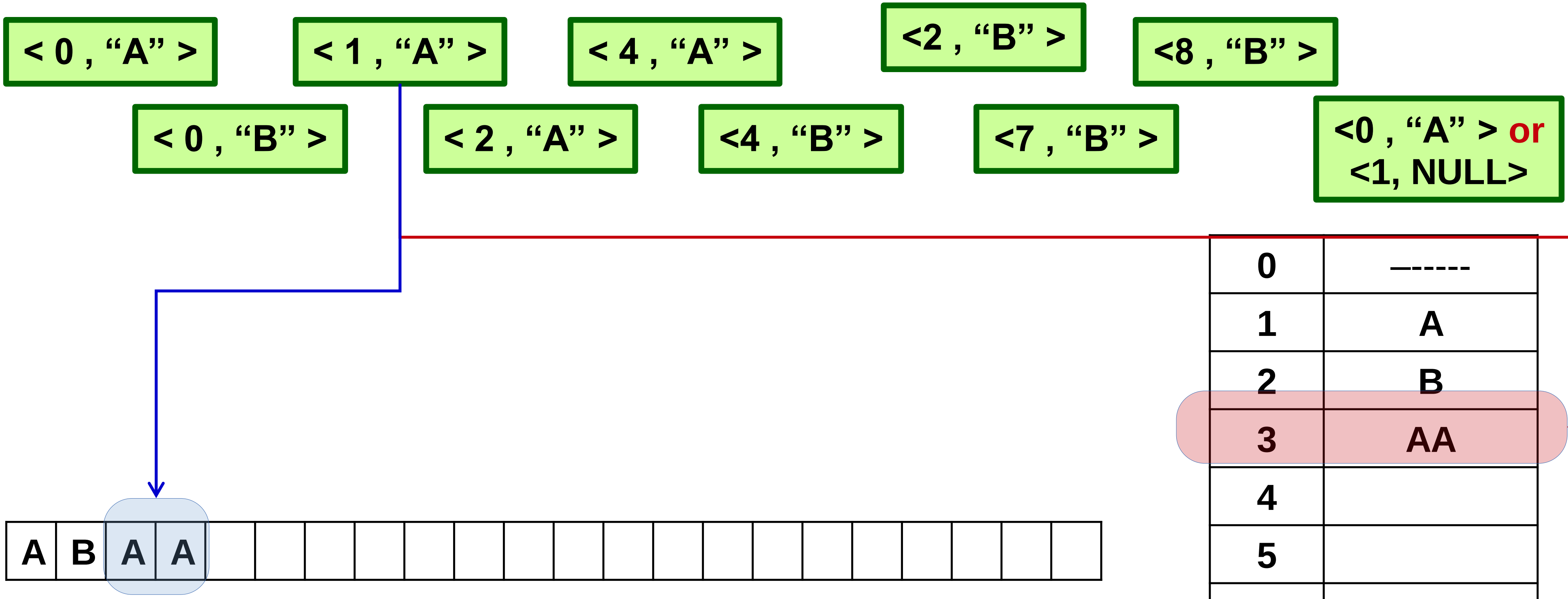
LZ 78 (De-Compression)



0	------
1	A
2	B
3	
4	
5	
6	
7	
8	
9	
10	
11	

Get symbol at Index [0] in Dictionary {""}
Concatenate Symbol "B" , Obtain {"B"}
Add {"B"} to Dictionary

LZ 78 (De-Compression)



< 0 , "A" >

< 0 , "B" >

< 1 , "A" >

< 2 , "A" >

< 4 , "A" >

<4 , "B" >

<2 , "B" >

<7 , "B" >

<8 , "B" >

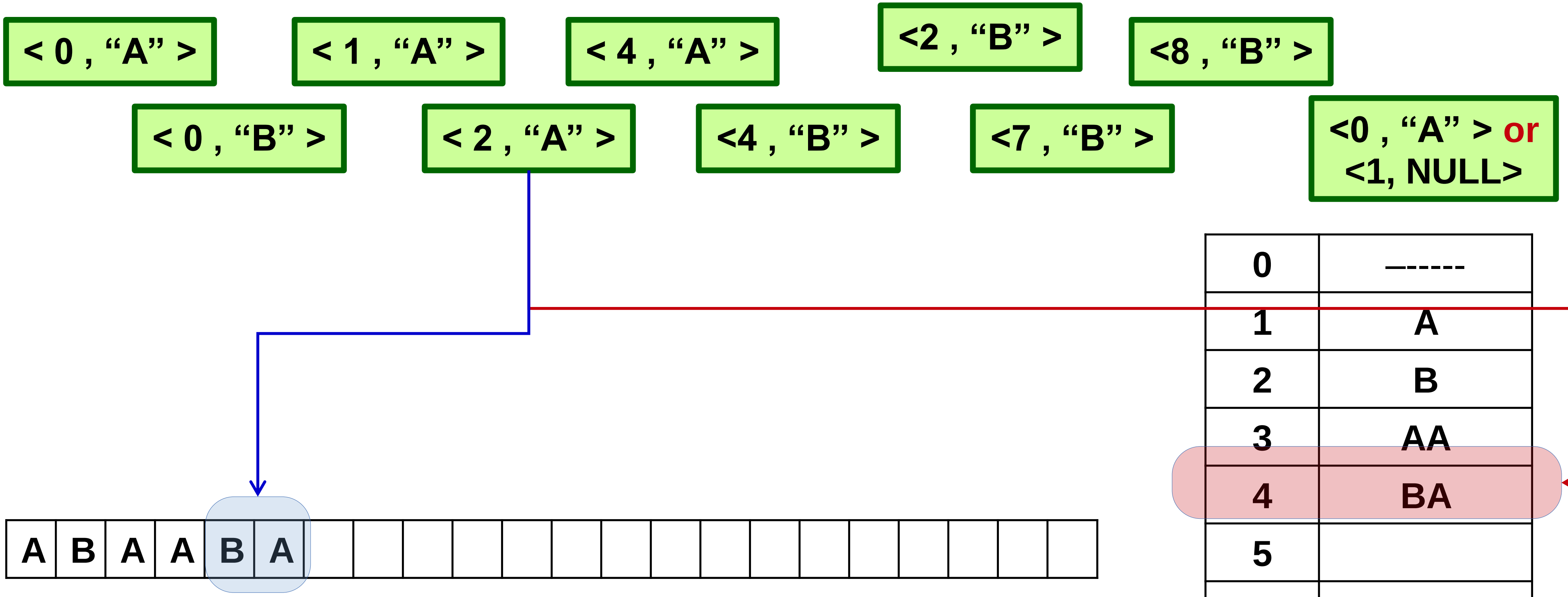
<0 , "A" > or <1, NULL>

A	B	A	A																		

0	------
1	A
2	B
3	AA
4	
5	
6	
7	
8	
9	
10	
11	

Get symbol at Index [1] in Dictionary {"A"}
Concatenate Symbol "B" , Obtain {"AA"}
Add {"AA"} to Dictionary

LZ 78 (De-Compression)



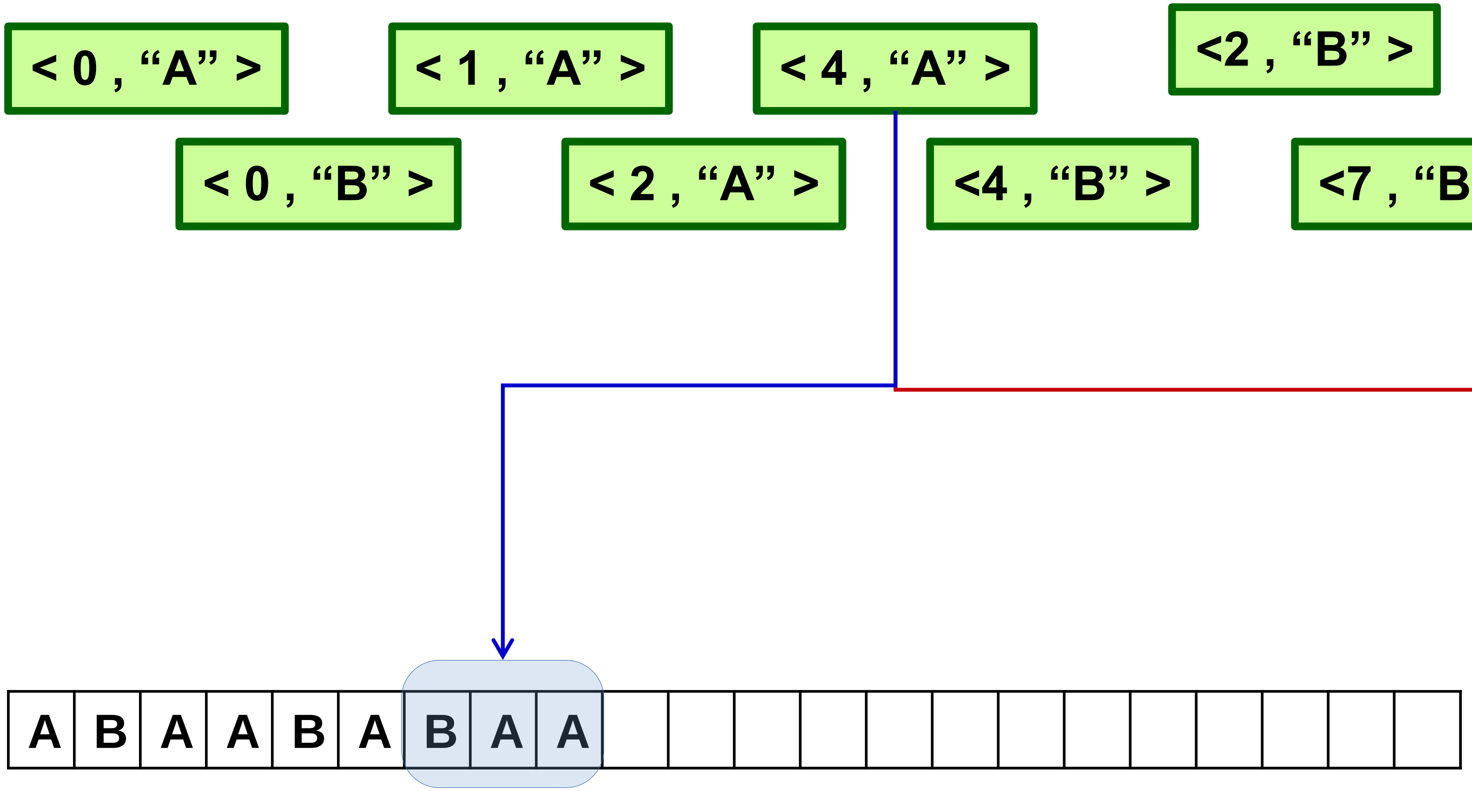
0	------
1	A
2	B
3	AA
4	BA
5	
6	
7	
8	
9	
10	
11	

Get symbol at Index [2] in Dictionary {“B”}

Concatenate Symbol “A” , Obtain {“BA”}

Add {“BA”} to Dictionary

LZ 78 (De-Compression)



< 0 , “A” >

< 0 , “B” >

< 1 , “A” >

< 2 , “A” >

< 4 , “A” >

<4 , “B” >

<2 , “B” >

<7 , “B” >

<8 , “B” >

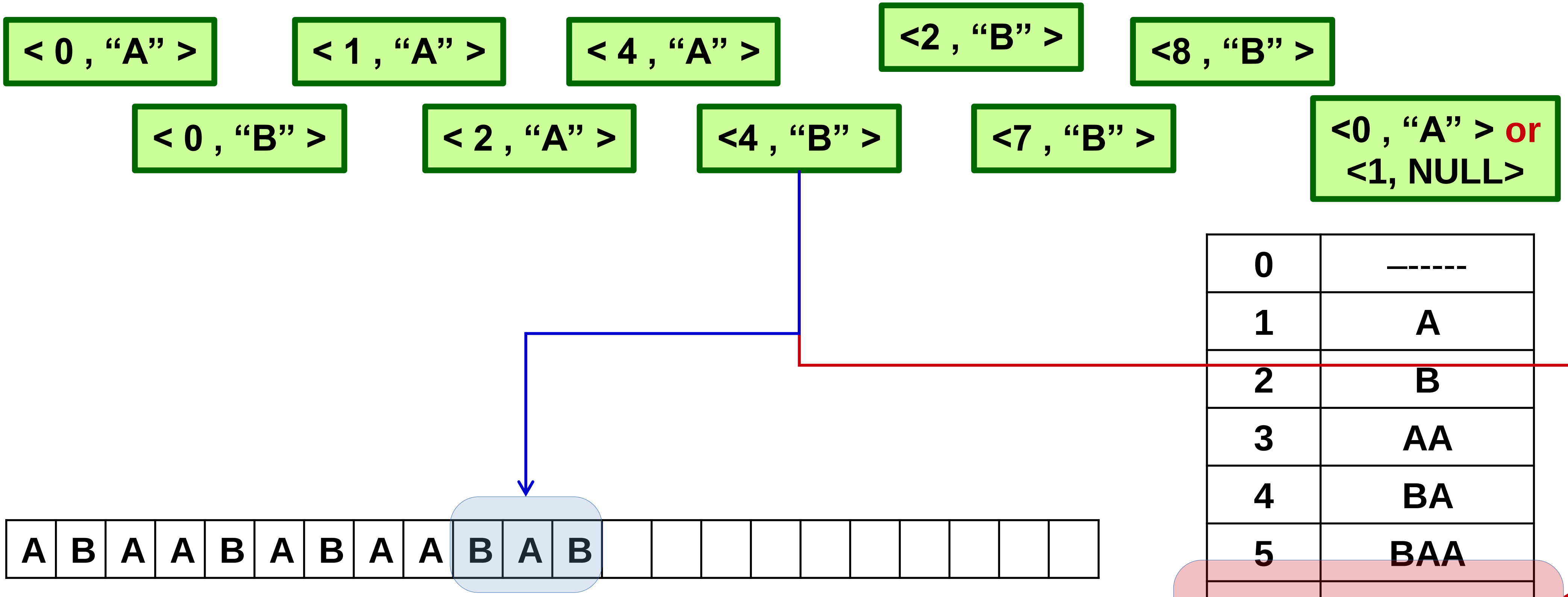
<0 , “A” > or <1, NULL>

A	B	A	A	B	A	B	A	A													

0	–-----
1	A
2	B
3	AA
4	BA
5	BAA
6	
7	
8	
9	
10	
11	

Get symbol at Index [4] in Dictionary {“BA”}
Concatenate Symbol “A” , Obtain {“BAA”}
Add {“BAA”} to Dictionary

LZ 78 (De-Compression)



< 0 , “A” >

< 0 , “B” >

< 1 , “A” >

< 2 , “A” >

< 4 , “A” >

<4 , “B” >

<2 , “B” >

<7 , “B” >

<8 , “B” >

<0 , “A” > or <1, NULL>

A	B	A	A	B	A	B	A	A	B	A	B										

Get symbol at Index [4] in Dictionary {“BA”}

Concatenate Symbol “B” , Obtain {“BAB”}

Add {“BAB”} to Dictionary

0	------
1	A
2	B
3	AA
4	BA
5	BAA
6	BAB
7	
8	
9	
10	
11	

LZ 78 (De-Compression)

< 0 , “A” >

< 0 , “B” >

< 1 , “A” >

< 2 , “A” >

< 4 , “A” >

<4 , “B” >

<2 , “B” >

<7 , “B” >

<8 , “B” >

<0 , “A” > or <1, NULL>

A	B	A	A	B	A	B	A	A	B	A	B	B	B								

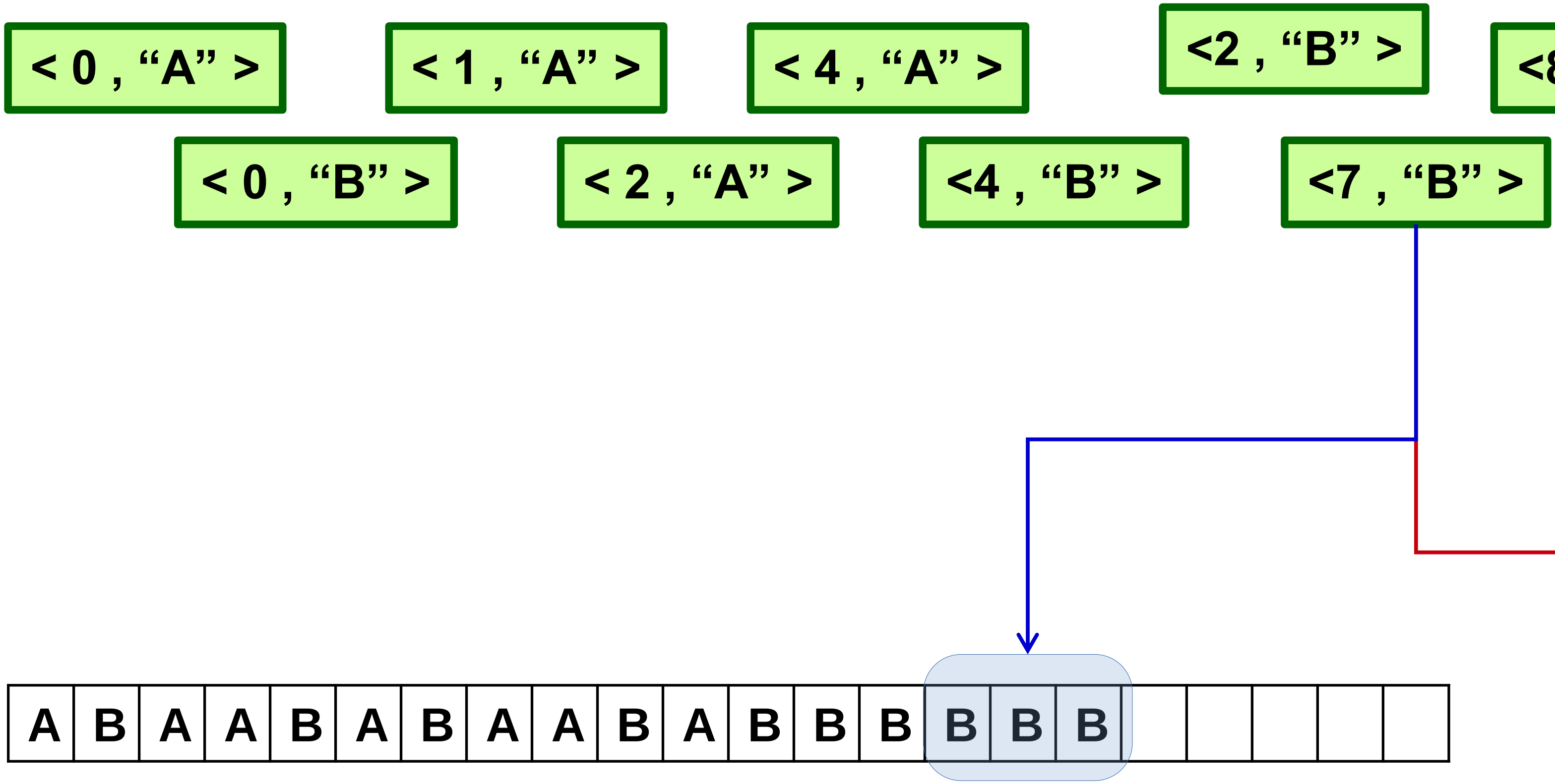
0	------
1	A
2	B
3	AA
4	BA
5	BAA
6	BAB
7	BB
8	
9	
10	
11	

Get symbol at Index [2] in Dictionary {“B”}

Concatenate Symbol “B” , Obtain {“BB”}

Add {“BB”} to Dictionary

LZ 78 (De-Compression)



< 0 , "A" > < 1 , "A" > < 4 , "A" > <2 , "B" > <8 , "B" >

< 0 , "B" > < 2 , "A" > <4 , "B" > <7 , "B" >

<0 , "A" > or <1, NULL>

A	B	A	A	B	A	B	A	A	B	A	B	B	B	B	B	B					

0	------
1	A
2	B
3	AA
4	BA
5	BAA
6	BAB
7	BB
8	BBB
9	
10	
11	

Get symbol at Index [7] in Dictionary {"BB"}
Concatenate Symbol "B" , Obtain {"BBB"}
Add {"BBB"} to Dictionary

LZ 78 (De-Compression)

< 0 , “A” >

< 0 , “B” >

< 1 , “A” >

< 2 , “A” >

< 4 , “A” >

<4 , “B” >

<2 , “B” >

<7 , “B” >

<8 , “B” >

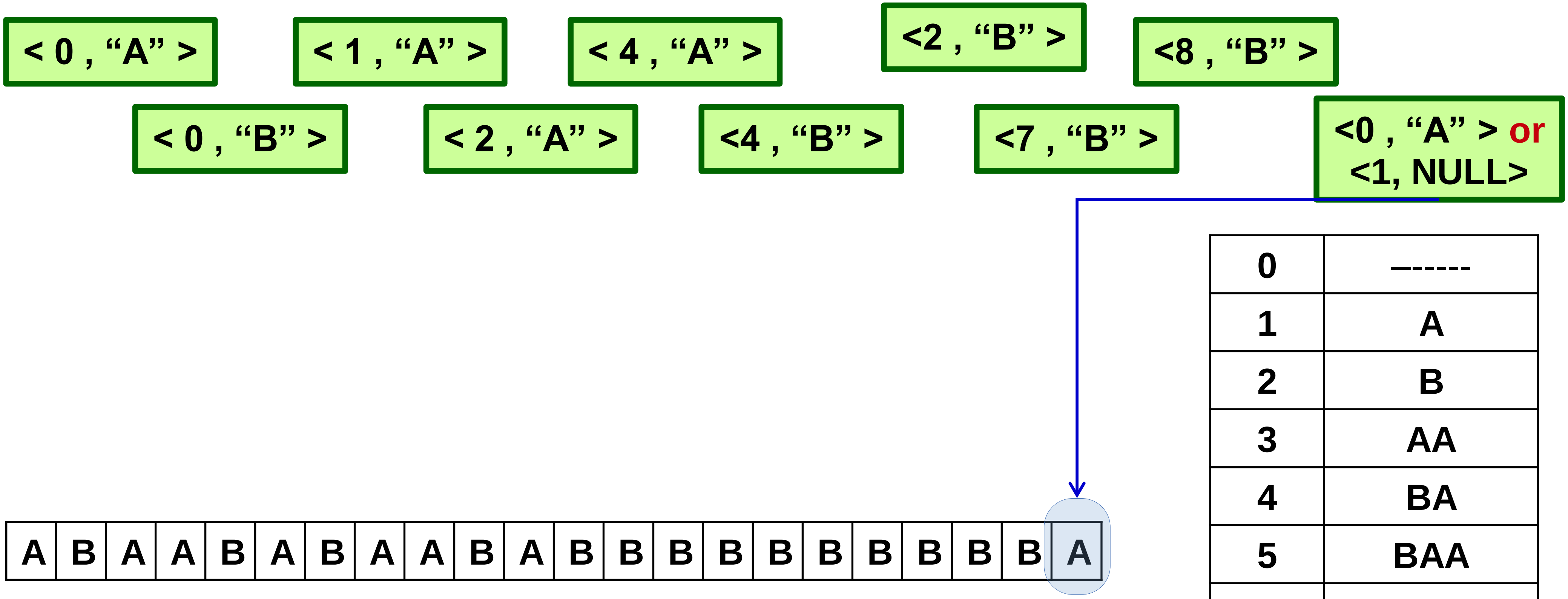
<0 , “A” > or <1, NULL>

A	B	A	A	B	A	B	A	A	B	A	B	B	B	B	B	B	B	B	B	B	

0	------
1	A
2	B
3	AA
4	BA
5	BAA
6	BAB
7	BB
8	BBB
9	BBBB
10	
11	

Get symbol at Index [8] in Dictionary {“BBB”}
Concatenate Symbol “B” , Obtain {“BBBB”}
Add {“BBBB”} to Dictionary

LZ 78 (De-Compression)



< 0 , “A” > < 1 , “A” > < 4 , “A” > <2 , “B” > <8 , “B” >

< 0 , “B” > < 2 , “A” > <4 , “B” > <7 , “B” > <0 , “A” > or <1, NULL>

A	B	A	A	B	A	B	A	A	B	A	B	B	B	B	B	B	B	B	B	B	A

0	------
1	A
2	B
3	AA
4	BA
5	BAA
6	BAB
7	BB
8	BBB
9	BBBB
10	
11	

Get symbol at Index [1] in Dictionary {“A”}
Concatenate Symbol “NULL” , Obtain {“A”}
Add NOTHING to Dictionary

LZ 78 Compression Ratio

< 0 , “A” >

< 0 , “B” >

< 1 , “A” >

< 2 , “A” >

< 4 , “A” >

<4 , “B” >

<2 , “B” >

<7 , “B” >

<8 , “B” >

<0 , “A” > or <1, NULL>

Original Size = Number of Symbols * Bits used to Store one Symbol
= 22 Symbols * 8 Bits / Symbol = 176 bits
(Store “Symbol” ASCII Code in 8 Bits)

Max “Index” Value = 8 — Store “Index” Value in 4 Bits
Max Symbols = 256 Symbol — Store “Symbol” ASCII Code in 8 Bits
Tag size = 4 + 8 =12 Bits

Number of Tags = 10 Tags
Compressed Size=10*12=120 bits

LZ 78: Main Features

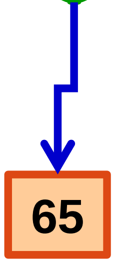
- No use of the sliding window.
- Instead of the triples used in the LZ77, only pairs are used in the LZ78. Specifically, only the **Position** (index in the list) of the matched string and the **Next Symbol** following the matched string need to be encoded (in the Tag).
- Use encoded text as a dictionary which, potentially, does not have a fixed size.
- Each time a Tag is issued, the encoded string is included in the dictionary.
- Once a preset limit to the dictionary size has been reached, it is reset to zero, i.e., it must be restarted.

LZW Compression

LZW compression works by reading a sequence of symbols, grouping the symbols into strings, and converting the strings into codes. Because the codes take up less space than the strings they replace, we get compression.

LZW Compression

A	B	A	A	B	A	B	B	A	A	B	A	A	B	A	A	A	A	B	A	B	B	B	B	B	B	B	B



65

...	...
65	A
66	B
...	...
...	...
128	AB
129	
130	
131	
132	
133	
134	
135	
136	
137	
138	
139	
140	
141	
142	

"A" exists in the table at index [65]

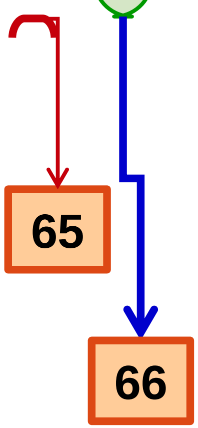
"AB" does NOT exist in the table

Save Symbol "A" as [65]

Add "AB" to Dictionary

LZW Compression

A	B	A	A	B	A	B	B	A	A	B	A	A	B	A	A	A	A	B	A	B	B	B	B	B	B	B	B



...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	
131	
132	
133	
134	
135	
136	
137	
138	
139	
140	
141	
142	

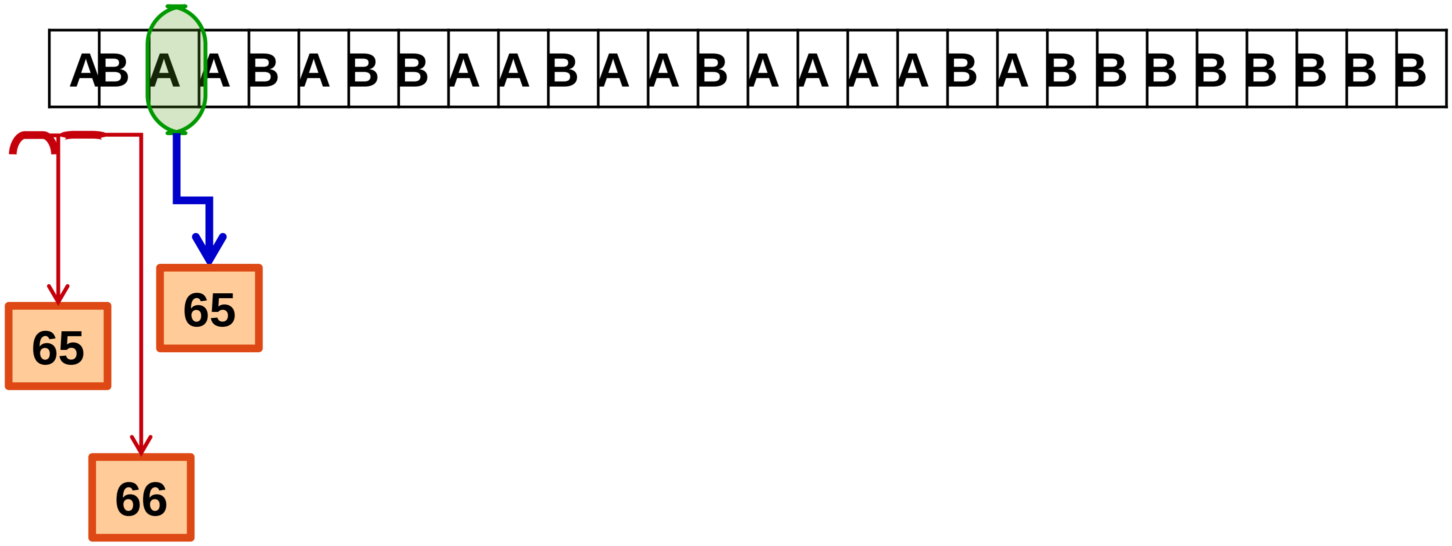
"B" exists in the table at index [66]

"BA" does NOT exist in the table

Save Symbol "B" as [66]

Add "BA" to Dictionary

LZW Compression



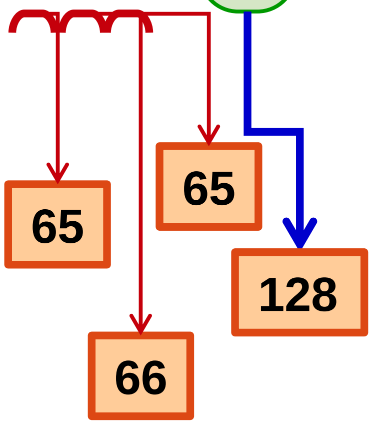
A	B	A	A	B	A	B	B	A	A	B	A	A	B	A	A	A	A	B	A	B	B	B	B	B	B	B	B

...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	
132	
133	
134	
135	
136	
137	
138	
139	
140	
141	
142	

"A" exists in the table at index [65]
"AA" does NOT exist in the table
Save Symbol "A" as [65]
Add "AA" to Dictionary

LZW Compression

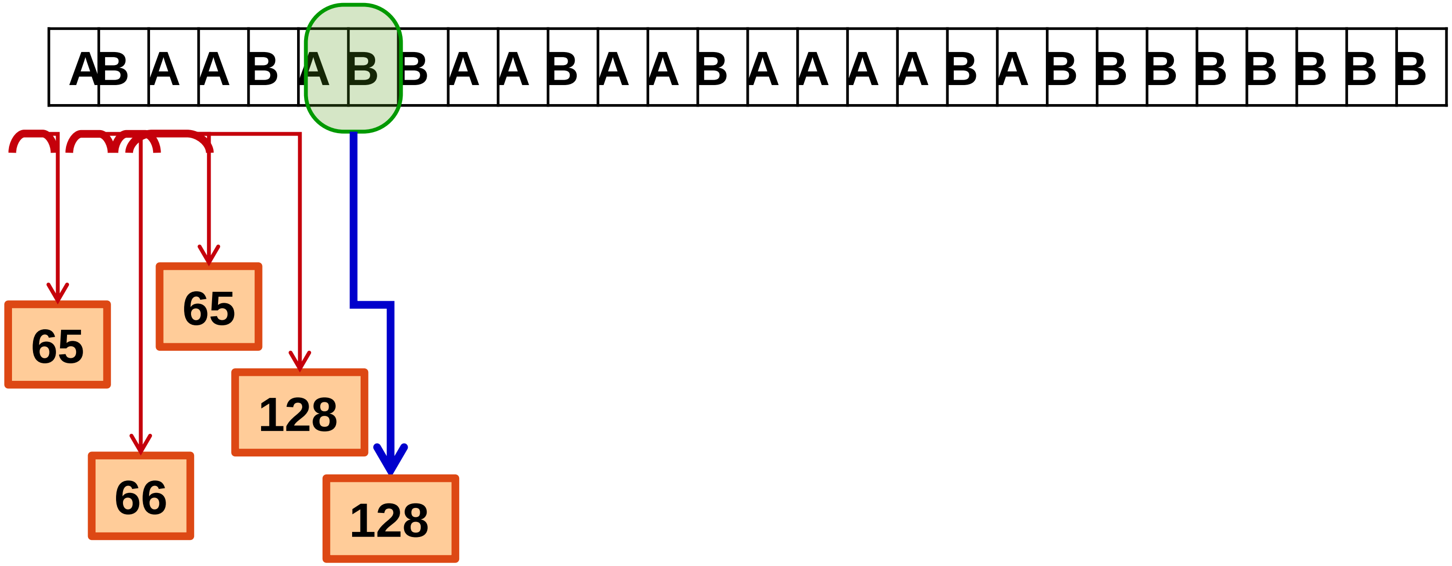
A	B	A	A	B	A	B	B	A	A	B	A	A	B	A	A	A	A	B	A	B	B	B	B	B	B	B	B



...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	
133	
134	
135	
136	
137	
138	
139	
140	
141	
142	

"AB" exists in the table at index [128]
"ABA" does NOT exist in the table
Save Symbol "AB" as [128]
Add "ABA" to Dictionary

LZW Compression



A	B	A	A	B	A	B	B	A	A	B	A	A	B	A	A	A	A	B	A	B	B	B	B	B	B	B	B

...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	
134	
135	
136	
137	
138	
139	
140	
141	
142	

"AB" exists in the table at index [128]

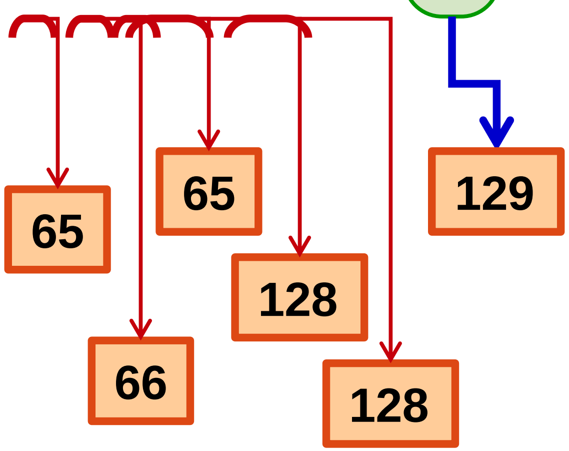
"ABB" does NOT exist in the table

Save Symbol "AB" as [128]

Add "ABB" to Dictionary

LZW Compression

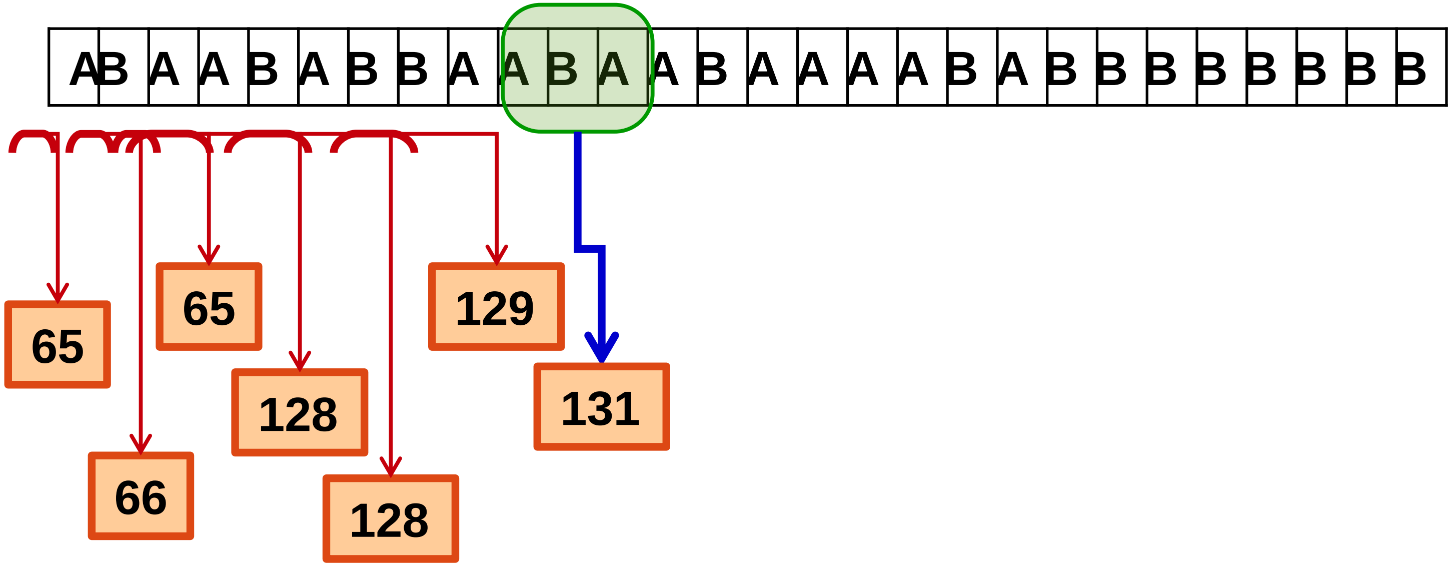
A	B	A	A	B	A	B	B	A	A	B	A	A	B	A	A	A	A	B	A	B	B	B	B	B	B	B	B



...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	
135	
136	
137	
138	
139	
140	
141	
142	

"BA" exists in the table at index [129]
"BAA" does NOT exist in the table
Save Symbol "BA" as [129]
Add "BAA" to Dictionary

LZW Compression

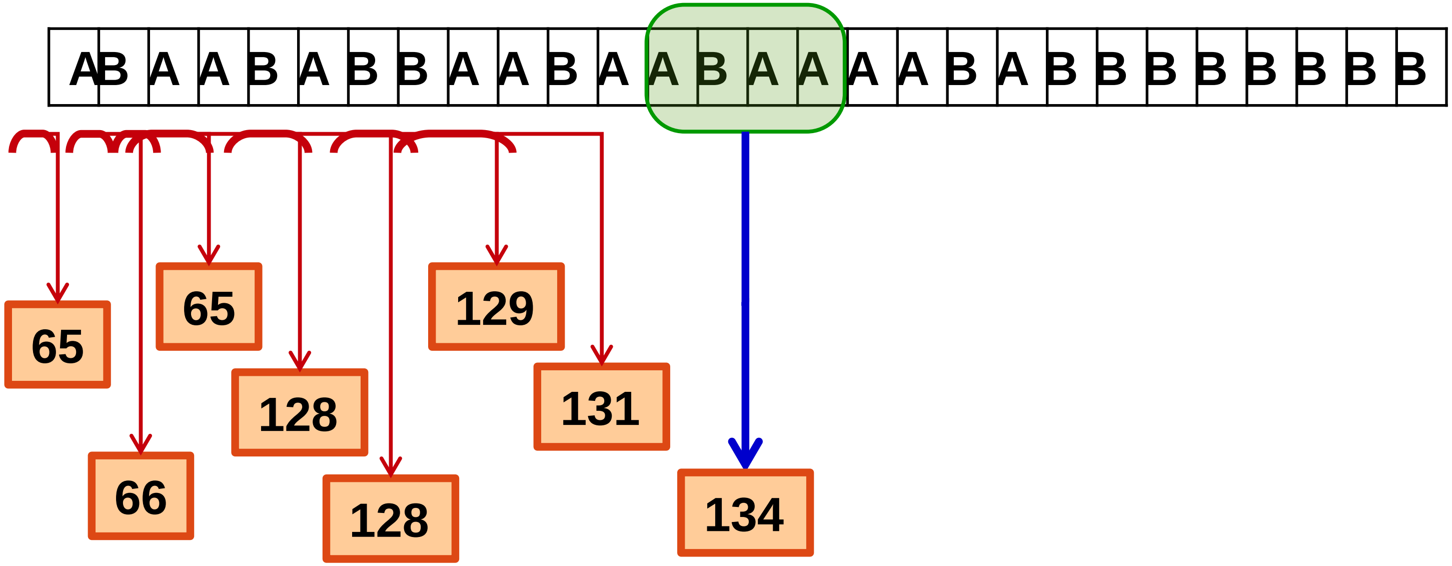


A B A A B A B B A A B A A B A A A A B A B B B B B B B B

...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	ABAA
135	
136	
137	
138	
139	
140	
141	
142	

"ABA" exists in the table at index [131]
"ABAA" does NOT exist in the table
Save Symbol "ABA" as [131]
Add "ABAA" to Dictionary

LZW Compression



...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	ABAA
135	ABAAA
136	
137	
138	
139	
140	
141	
142	

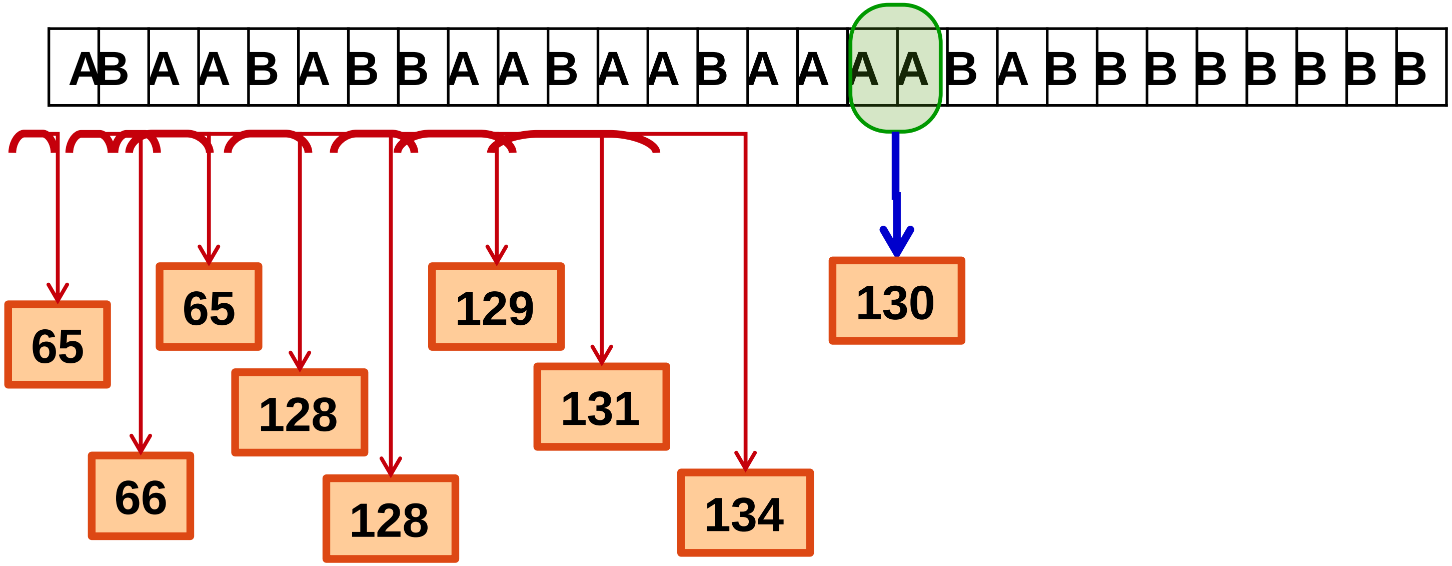
“ABAA” exists in the table at index [134]

“ABAAA” does NOT exist in the table

Save Symbol “ABAA” as [134]

Add “ABAAA” to Dictionary

LZW Compression



...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	ABAA
135	ABAAA
136	AAB
137	
138	
139	
140	
141	
142	

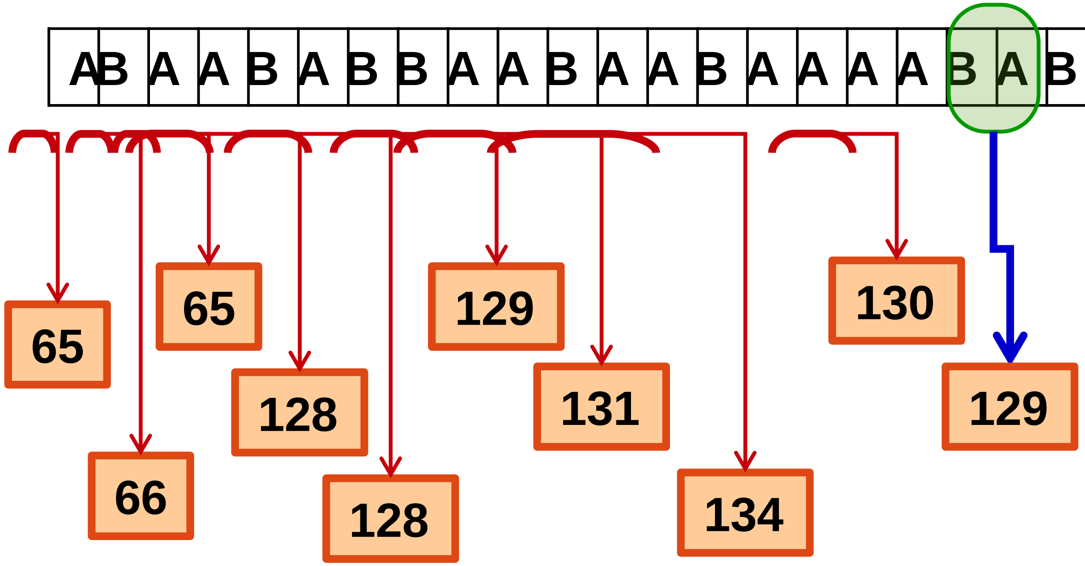
"AA" exists in the table at index [130]

"AAB" does NOT exist in the table

Save Symbol "AA" as [130]

Add "AAB" to Dictionary

LZW Compression



...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	ABAA
135	ABAAA
136	AAB
137	BAB
138	
139	
140	
141	
142	

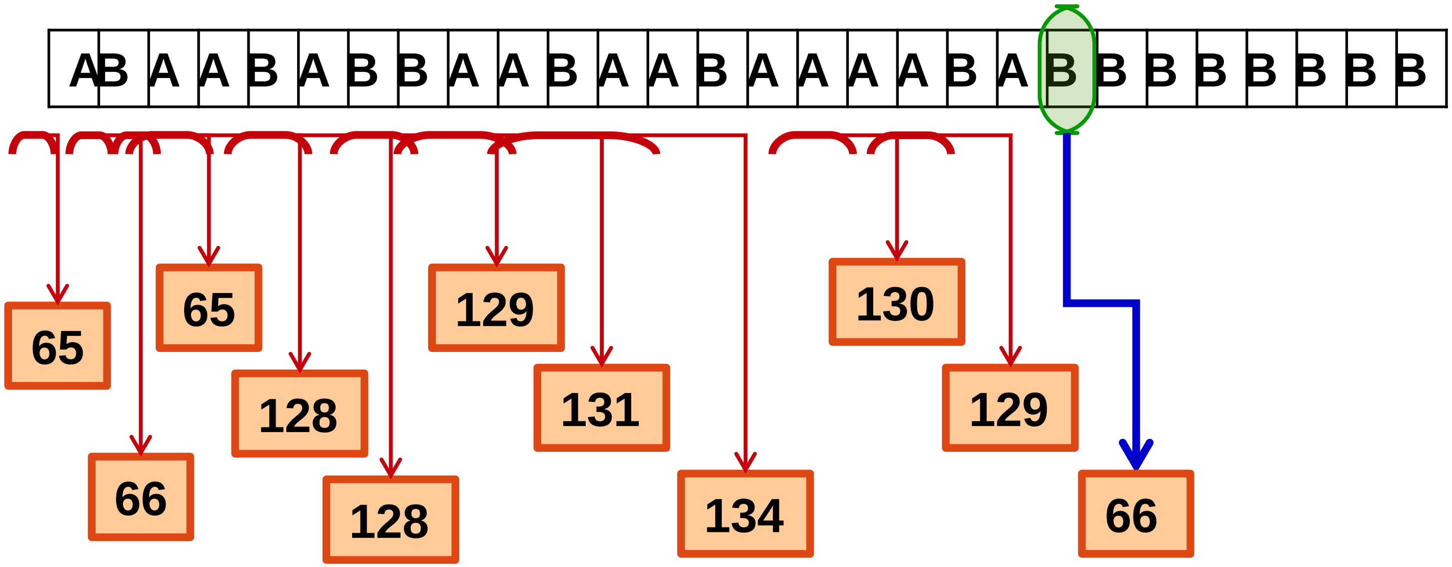
"BA" exists in the table at index [129]

"BAB" does NOT exist in the table

Save Symbol "BA" as [129]

Add "BAB" to Dictionary

LZW Compression



...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	ABAA
135	ABAAA
136	AAB
137	BAB
138	BB
139	
140	
141	
142	

"B" exists in the table at index [66]

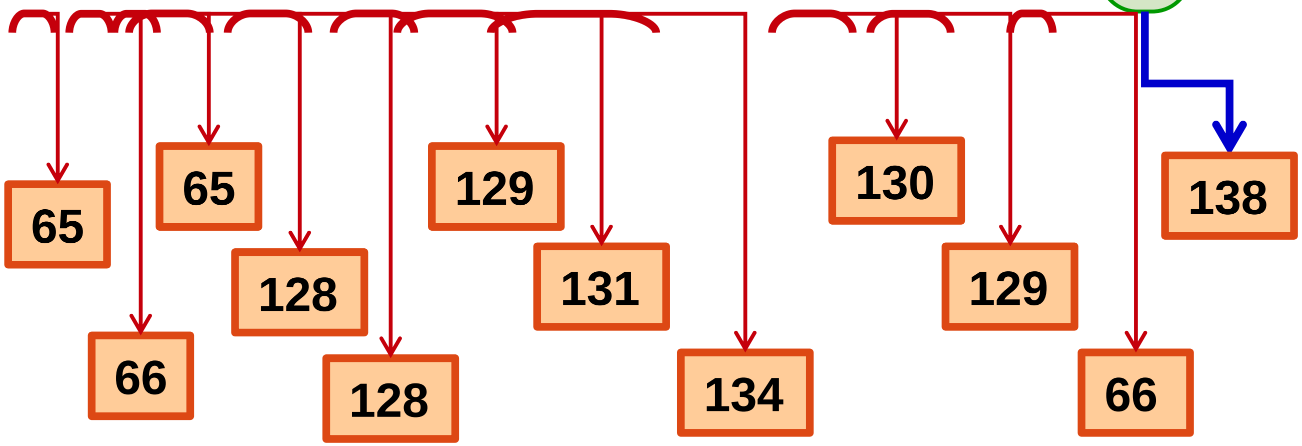
"BB" does NOT exist in the table

Save Symbol "B" as [66]

Add "BB" to Dictionary

LZW Compression

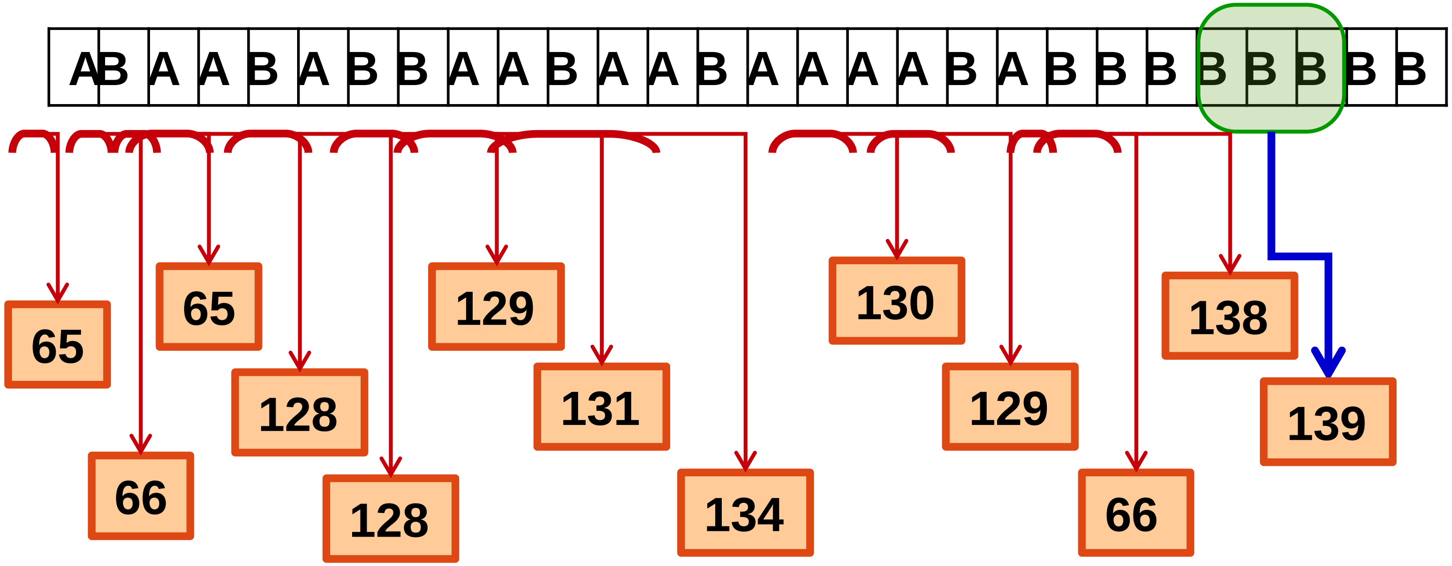
A	B	A	A	B	A	B	B	A	A	B	A	A	B	A	A	A	A	B	A	B	B	B	B	B	B	B	B



...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	ABAA
135	ABAAA
136	AAB
137	BAB
138	BB
139	BBB
140	
141	
142	

“BB” exists in the table at index [138]
“BBB” does NOT exist in the table
Save Symbol “BB” as [138]
Add “BBB” to Dictionary

LZW Compression

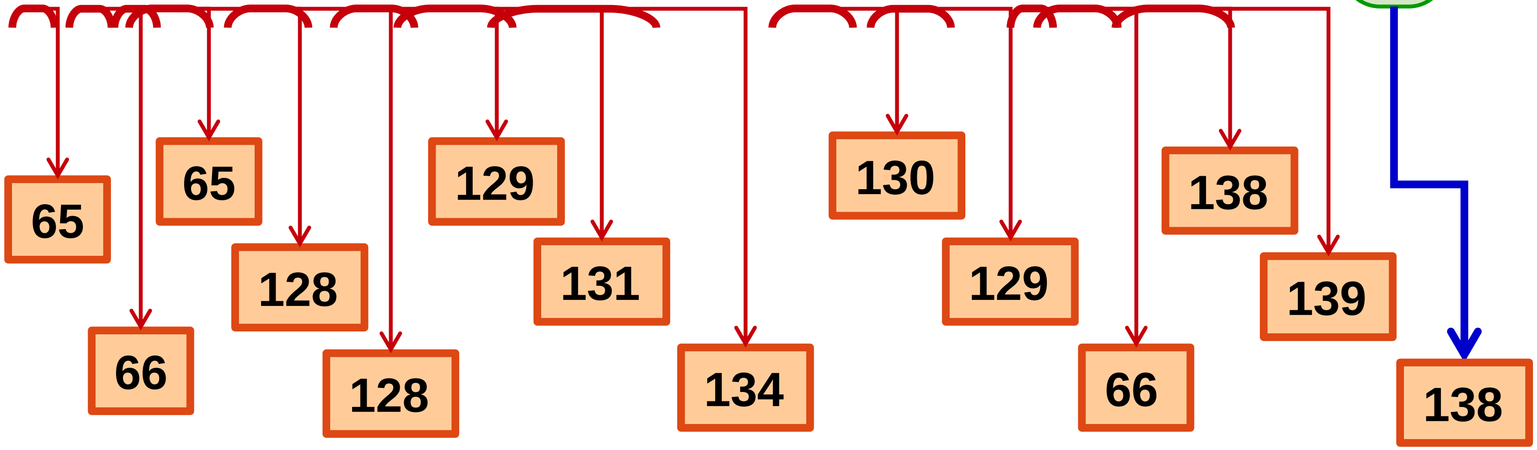


"BBB" exists in the table at index [139]
"BBBB" does NOT exist in the table
Save Symbol "BBB" as [139]
Add "BBBB" to Dictionary

...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	ABAA
135	ABAAA
136	AAB
137	BAB
138	BB
139	BBB
140	BBBB
141	
142	

LZW Compression

A B A A B A B B A A B A A B A A A A B A B B B B B B B B



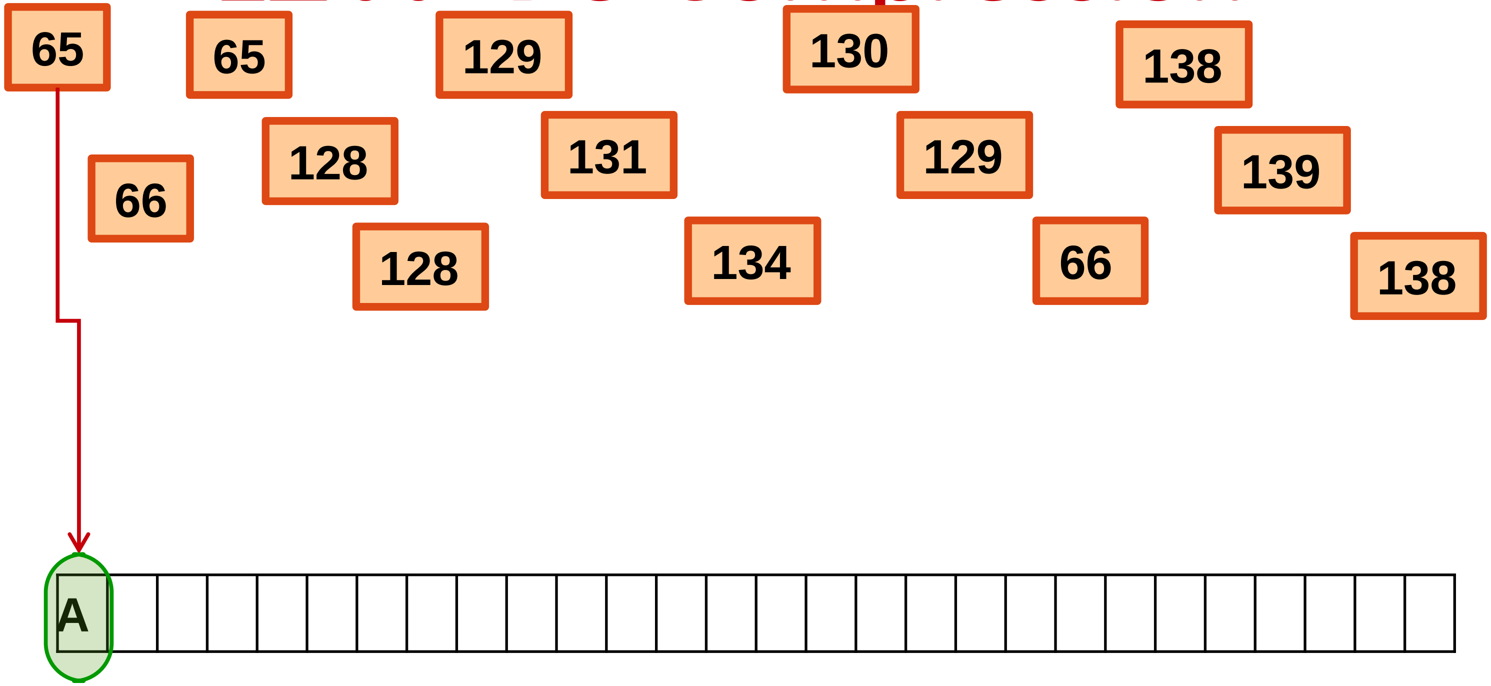
“BB” exists in the table at index [138]

Save Symbol “BB” as [138]

Add NOTHING to Dictionary

...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	ABAA
135	ABAAA
136	AAB
137	BAB
138	BB
139	BBB
140	BBBB
141	
142	

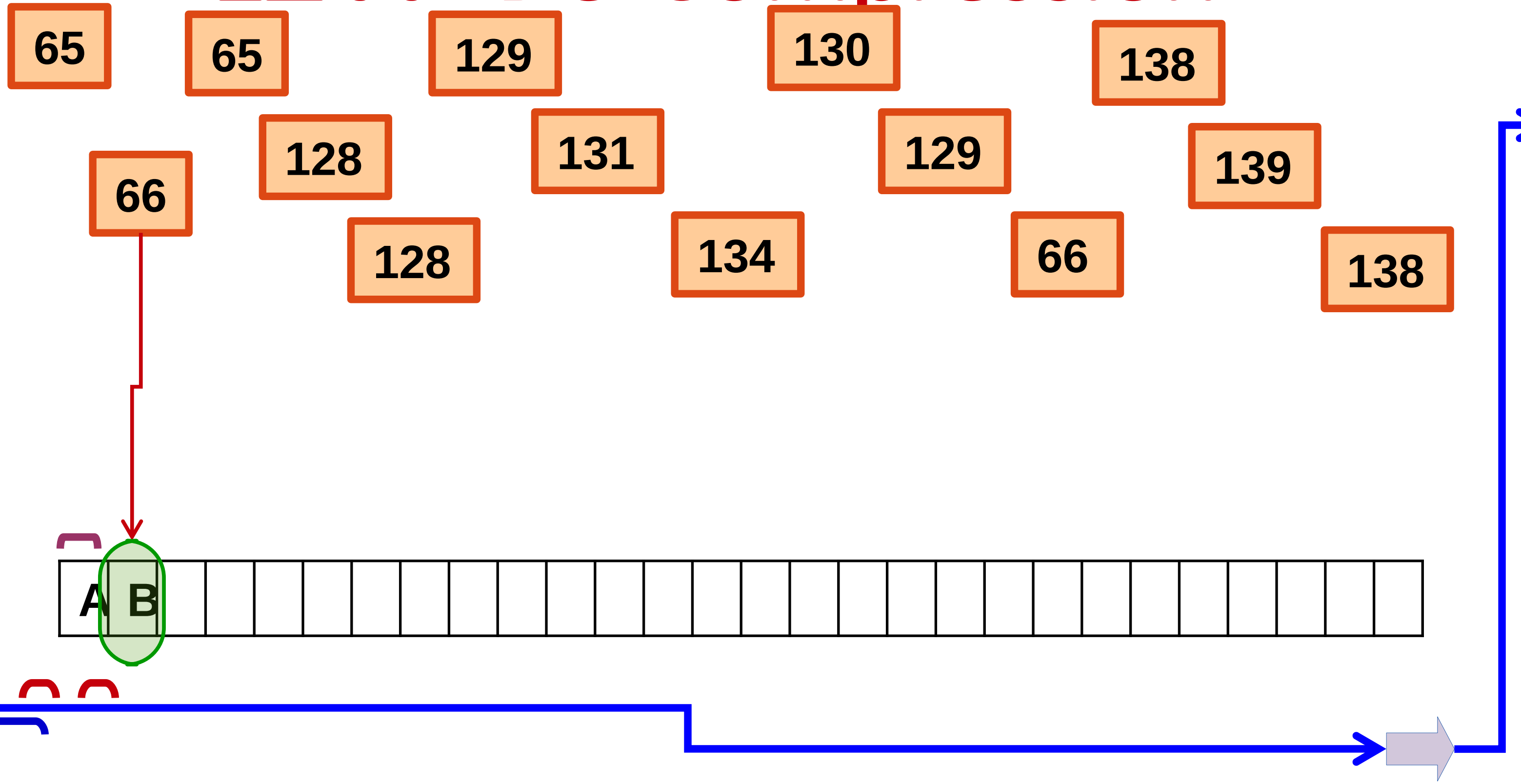
LZW De-Compression



...	...
65	A
66	B
...	...
...	...
128	
129	
130	
131	
132	
133	
134	
135	
136	
137	
138	
139	
140	
141	
142	

Pick symbol at Index [65] from the Dictionary; "A"
Add __NOTHING__ to Dictionary
(as this is the first symbol)

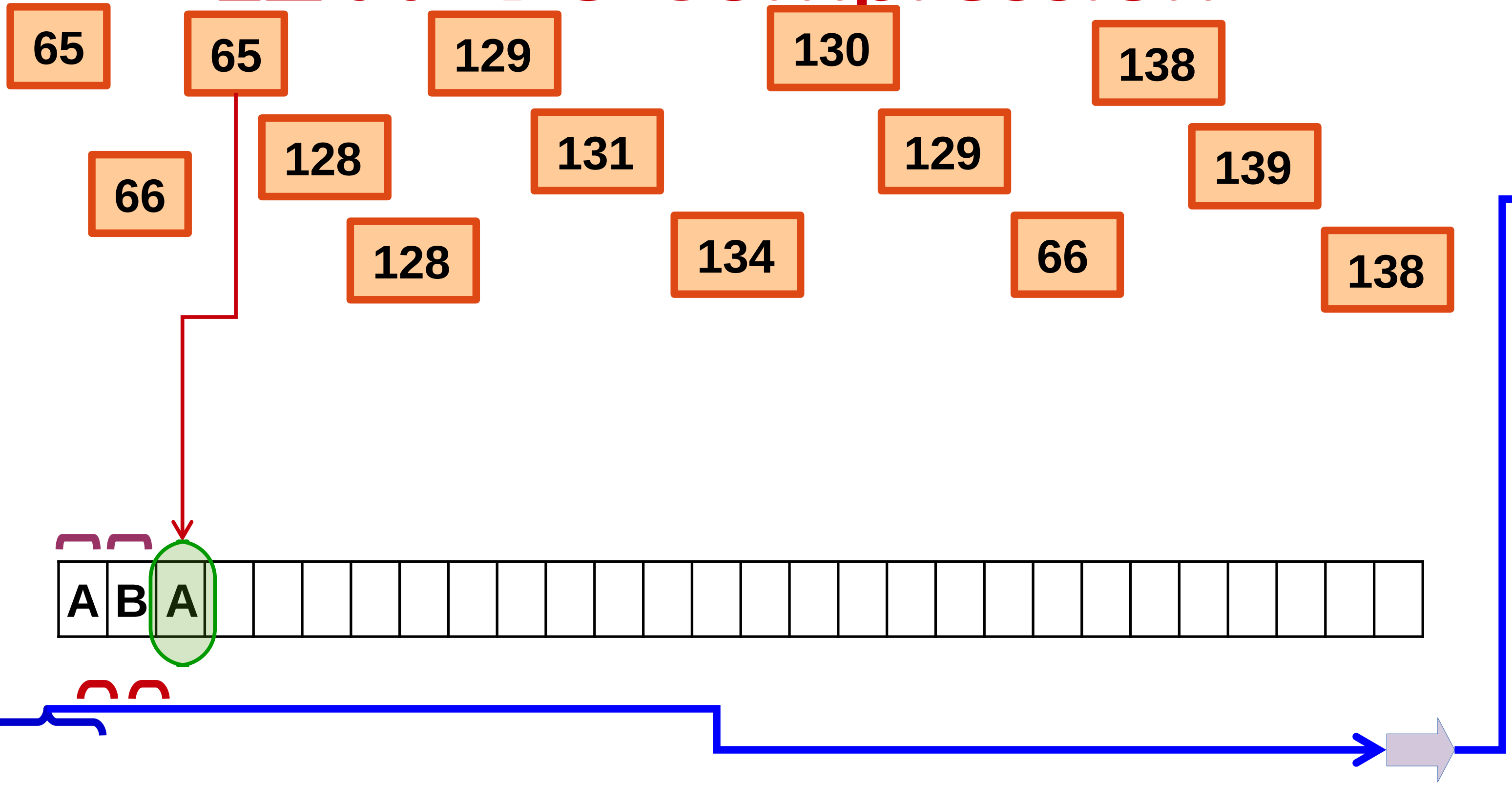
LZW De-Compression



...	...
65	A
66	B
...	...
...	...
128	AB
129	
130	
131	
132	
133	
134	
135	
136	
137	
138	
139	
140	
141	
142	

Pick symbols at Index [66] from the Dictionary; "B"
Concatenate ***ALL*** **Symbols picked from** ***Previous step***
and ***first*** **Symbol picked from** ***current step***
(Add concatenated Symbols to Dictionary)

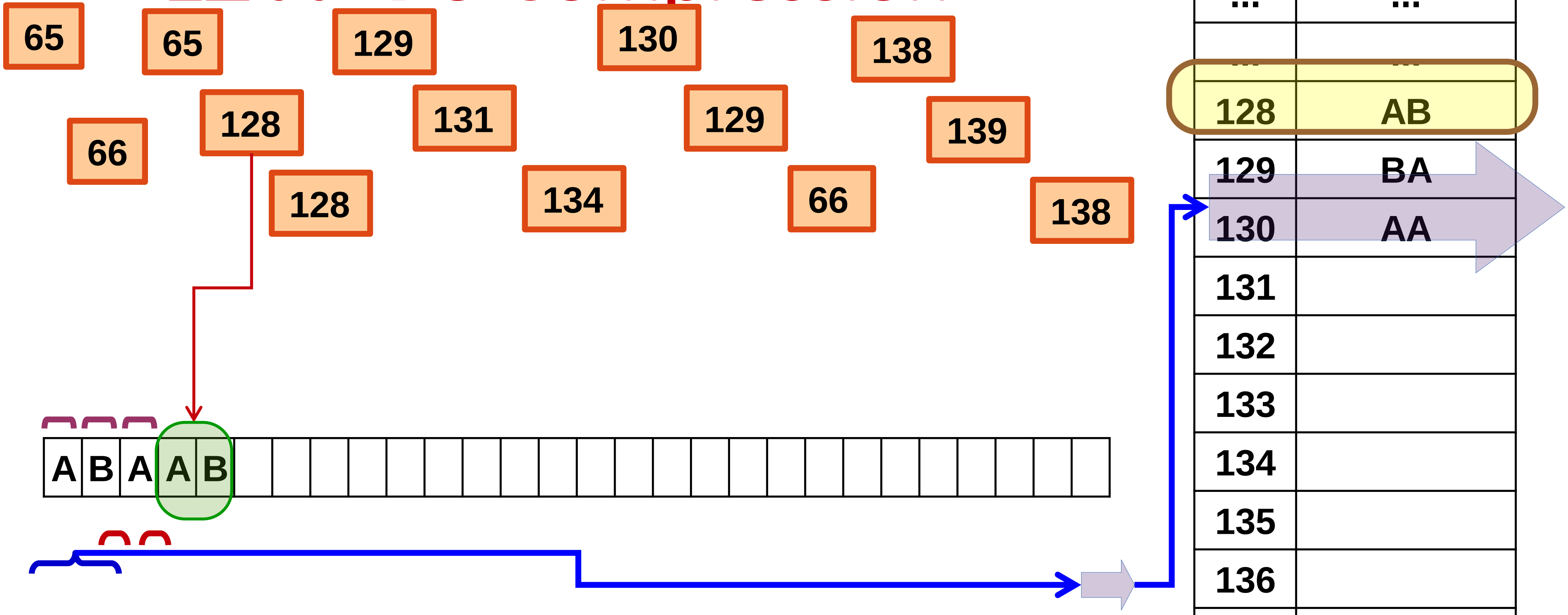
LZW De-Compression



...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	
131	
132	
133	
134	
135	
136	
137	
138	
139	
140	
141	
142	

Pick symbols at Index [65] from the Dictionary; "A"
Concatenate ALL Symbols picked from Previous step
and first Symbol picked from current step
(Add concatenated Symbols to Dictionary)

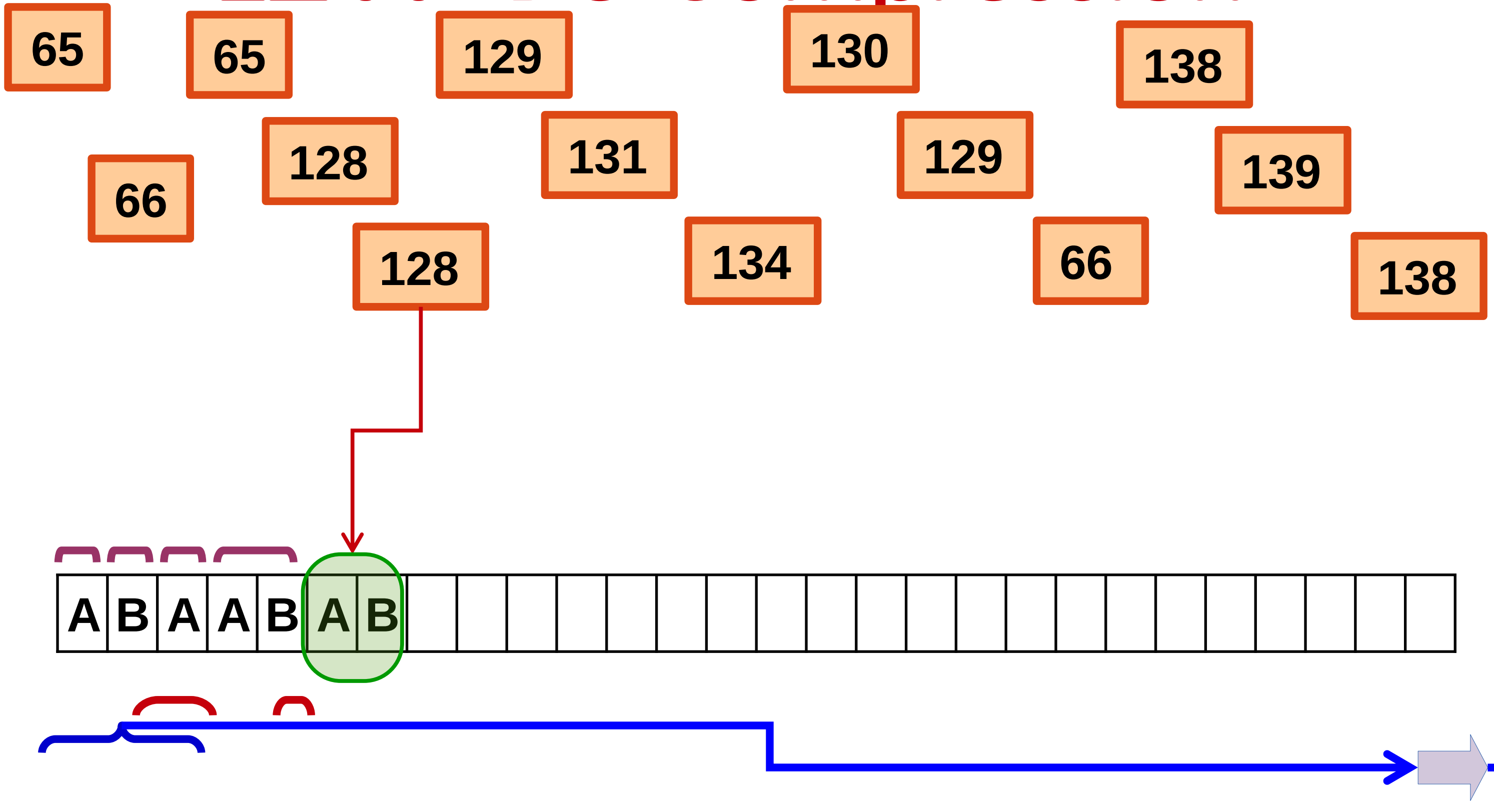
LZW De-Compression



...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	
132	
133	
134	
135	
136	
137	
138	
139	
140	
141	
142	

Pick symbols at Index [128] from the Dictionary; "AB"
Concatenate ALL Symbols picked from Previous step
and first Symbol picked from current step
(Add concatenated Symbols to Dictionary)

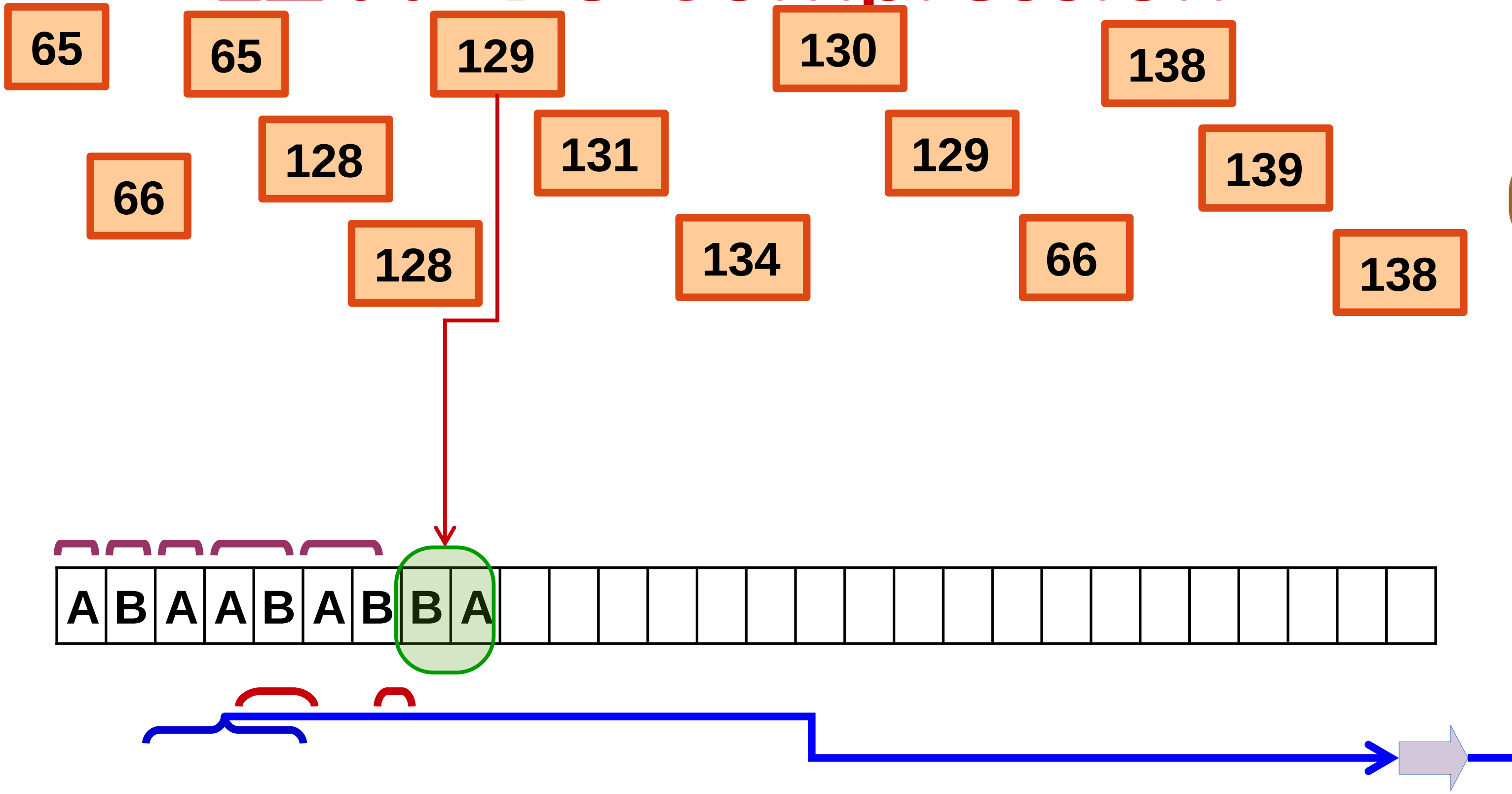
LZW De-Compression



...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	
133	
134	
135	
136	
137	
138	
139	
140	
141	
142	

Pick symbols at Index [128] from the Dictionary; "AB"
Concatenate ***ALL*** **Symbols picked from** ***Previous step***
and ***first*** **Symbol picked from** ***current step***
(Add concatenated ***Symbols*** *to Dictionary)*

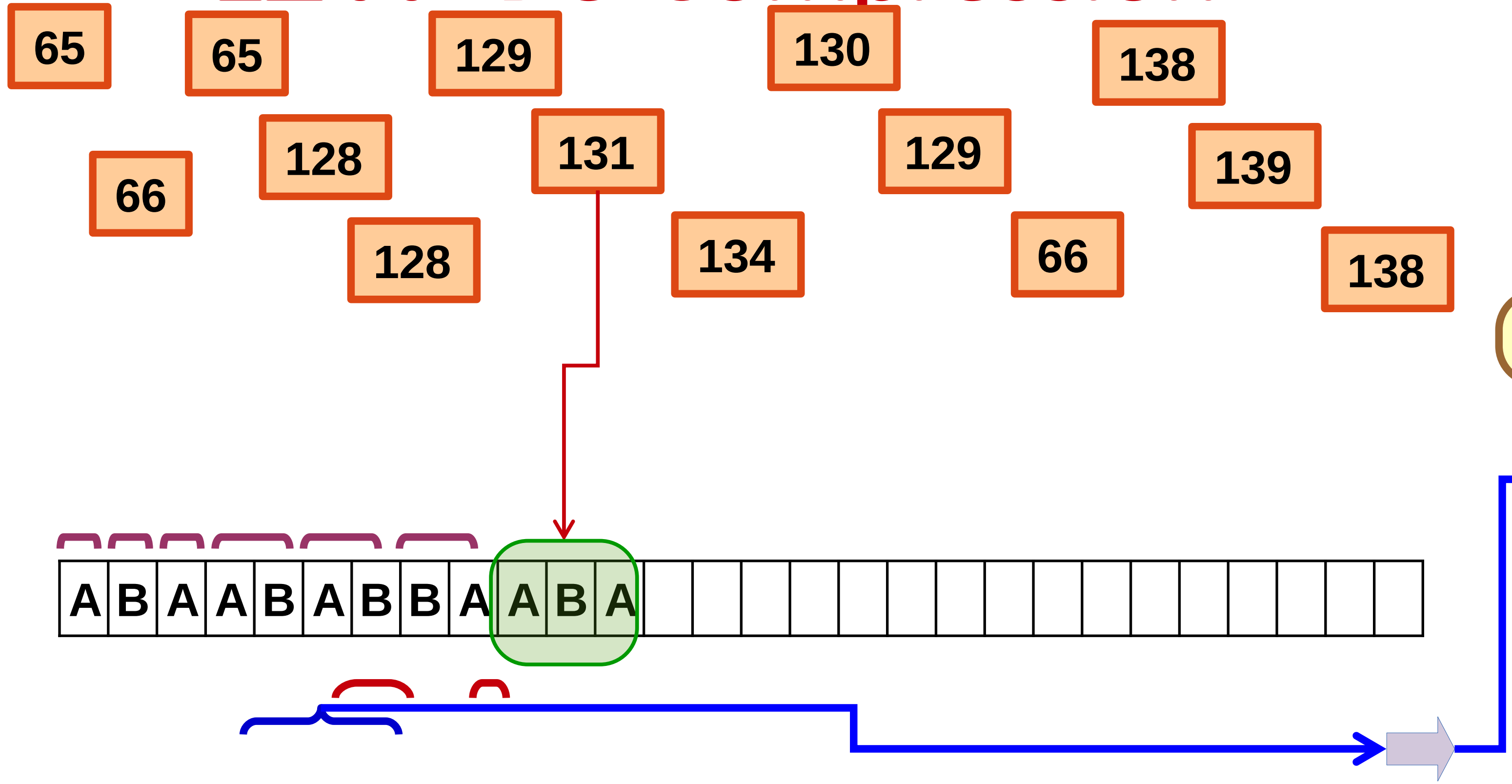
LZW De-Compression



...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	
134	
135	
136	
137	
138	
139	
140	
141	
142	

Pick symbols at Index [129] from the Dictionary; "BA"
Concatenate ALL Symbols picked from Previous step
and first Symbol picked from current step
(Add concatenated Symbols to Dictionary)

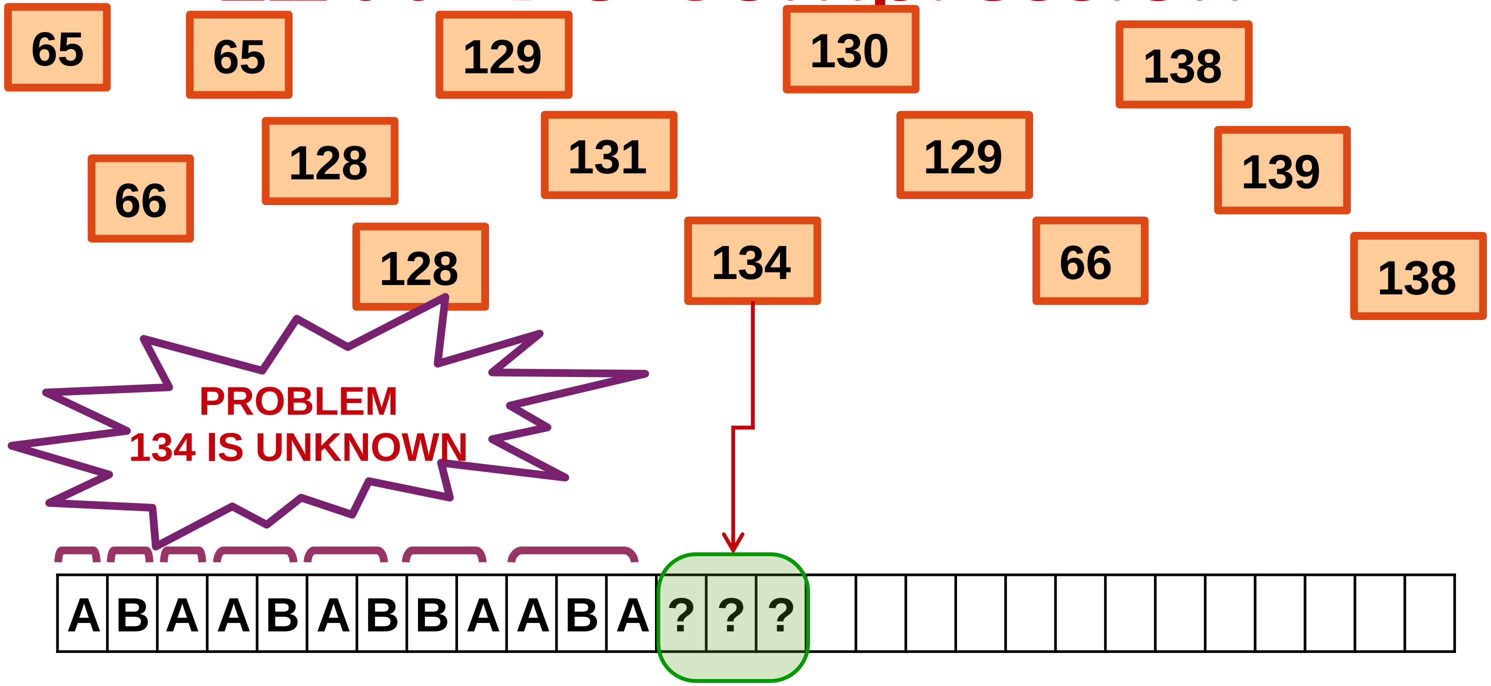
LZW De-Compression



...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	
135	
136	
137	
138	
139	
140	
141	
142	

Pick symbols at Index [131] from the Dictionary; "ABA"
Concatenate ALL Symbols picked from Previous step
and first Symbol picked from current step
(Add concatenated Symbols to Dictionary)

LZW De-Compression



...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	???
135	
136	
137	
138	
139	
140	
141	
142	

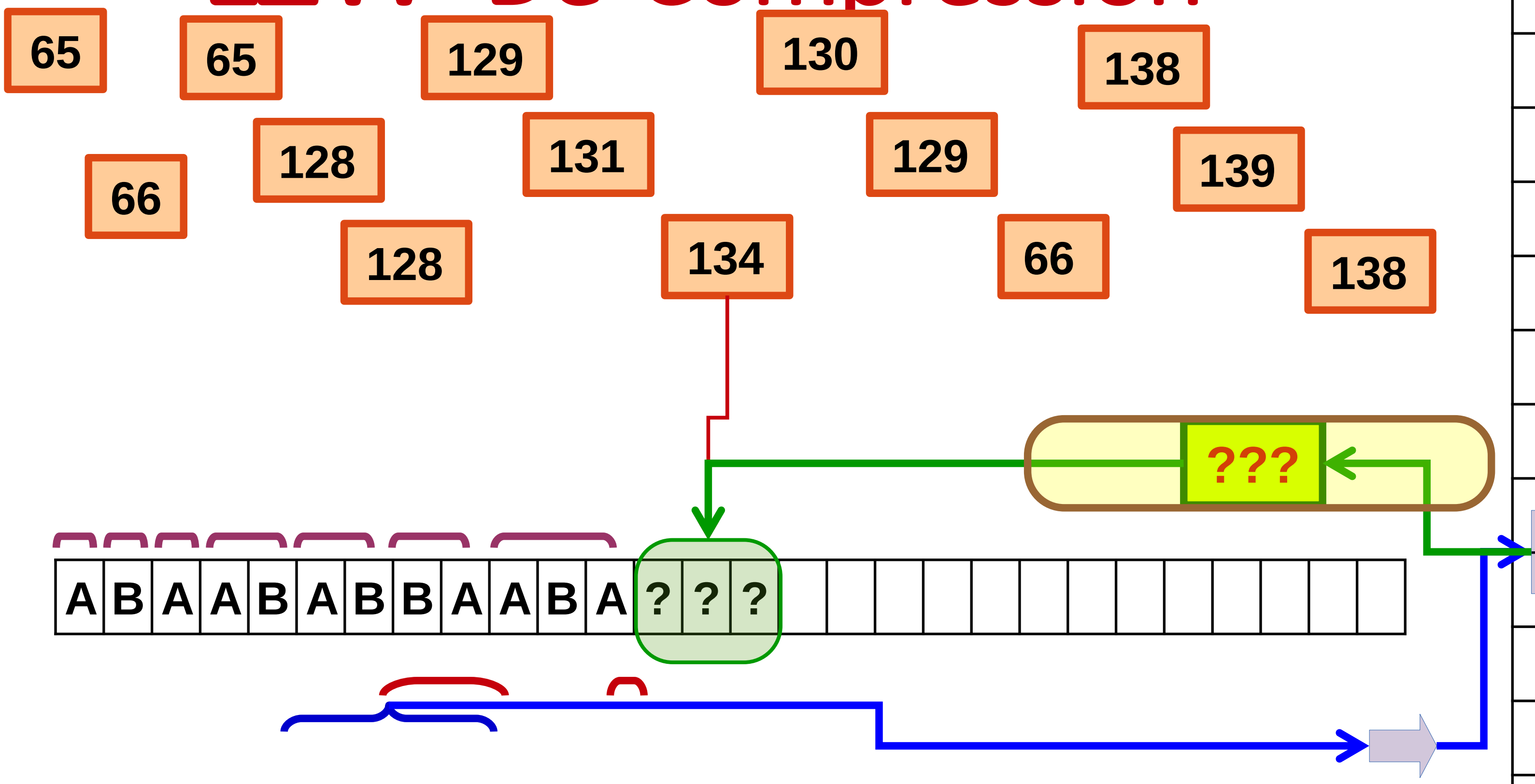
Pick symbols at Index [134] from the Dictionary

Symbols at Index [134] are not constructed yet

Assume they are "???" for the time being

Continue the algorithm

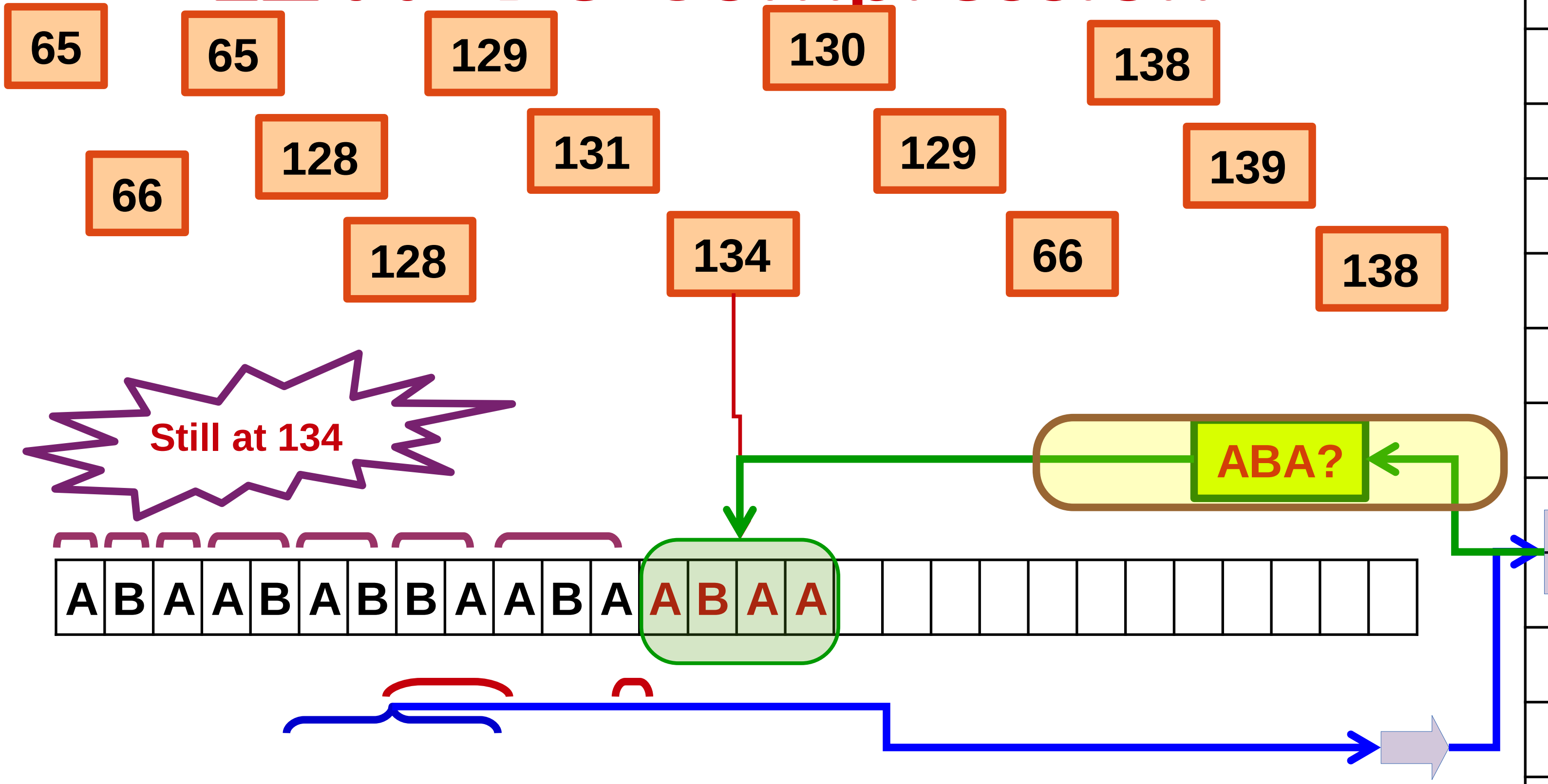
LZW De-Compression



…	…
65	A
66	B
…	…
…	…
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	ABA?
135	
136	
137	
138	
139	
140	
141	
142	

Pick symbols at Index [134] from the Dictionary; "???"
Concatenate ALL Symbols picked from Previous step
and first Symbol picked from current step
(Add concatenated Symbols to Dictionary)

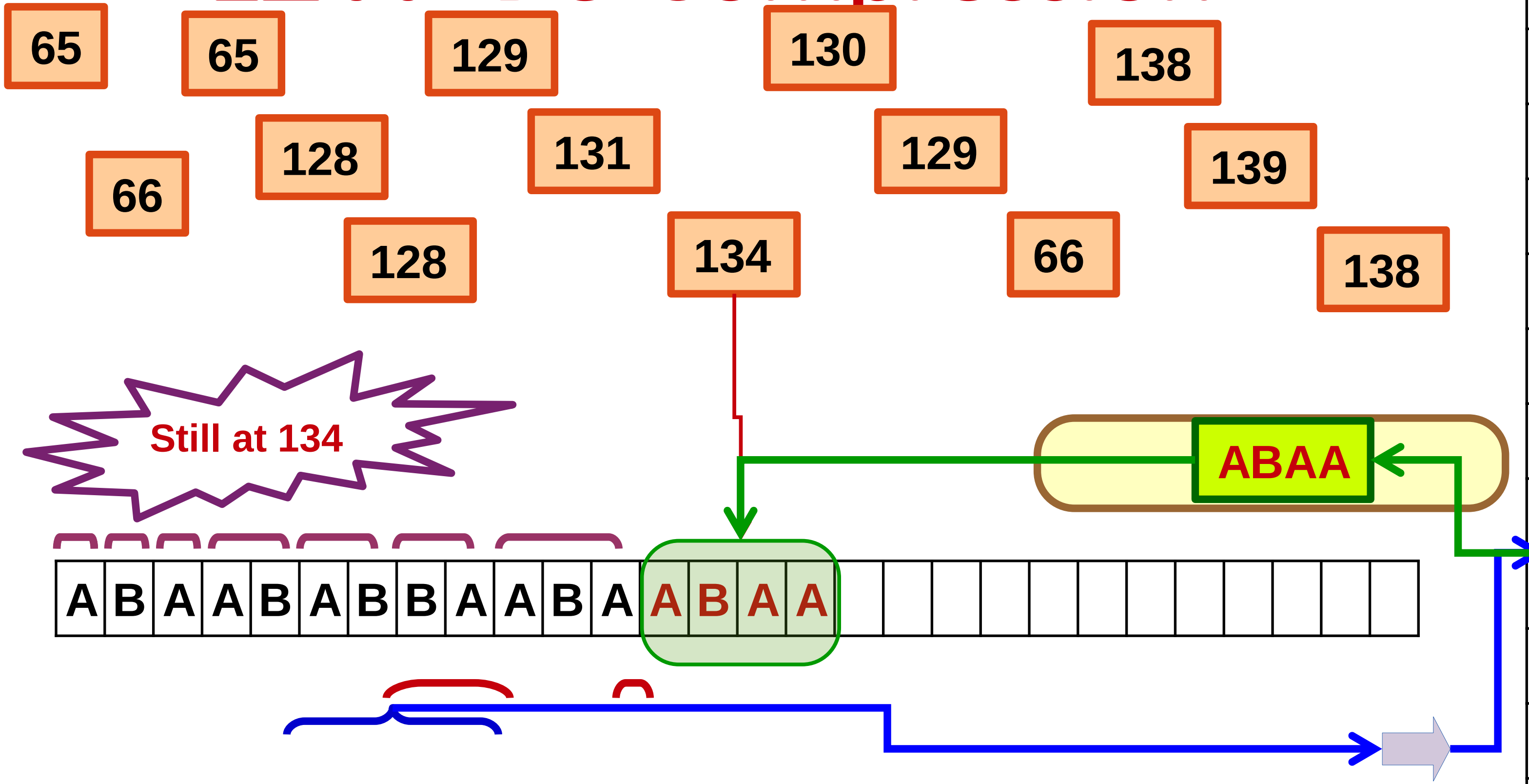
LZW De-Compression



...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	ABAA
135	
136	
137	
138	
139	
140	
141	
142	

Pick symbols at Index [134] from the Dictionary; "ABA?"
Concatenate ALL Symbols picked from Previous step
and first Symbol picked from current step
(Add concatenated Symbols to Dictionary)

LZW De-Compression

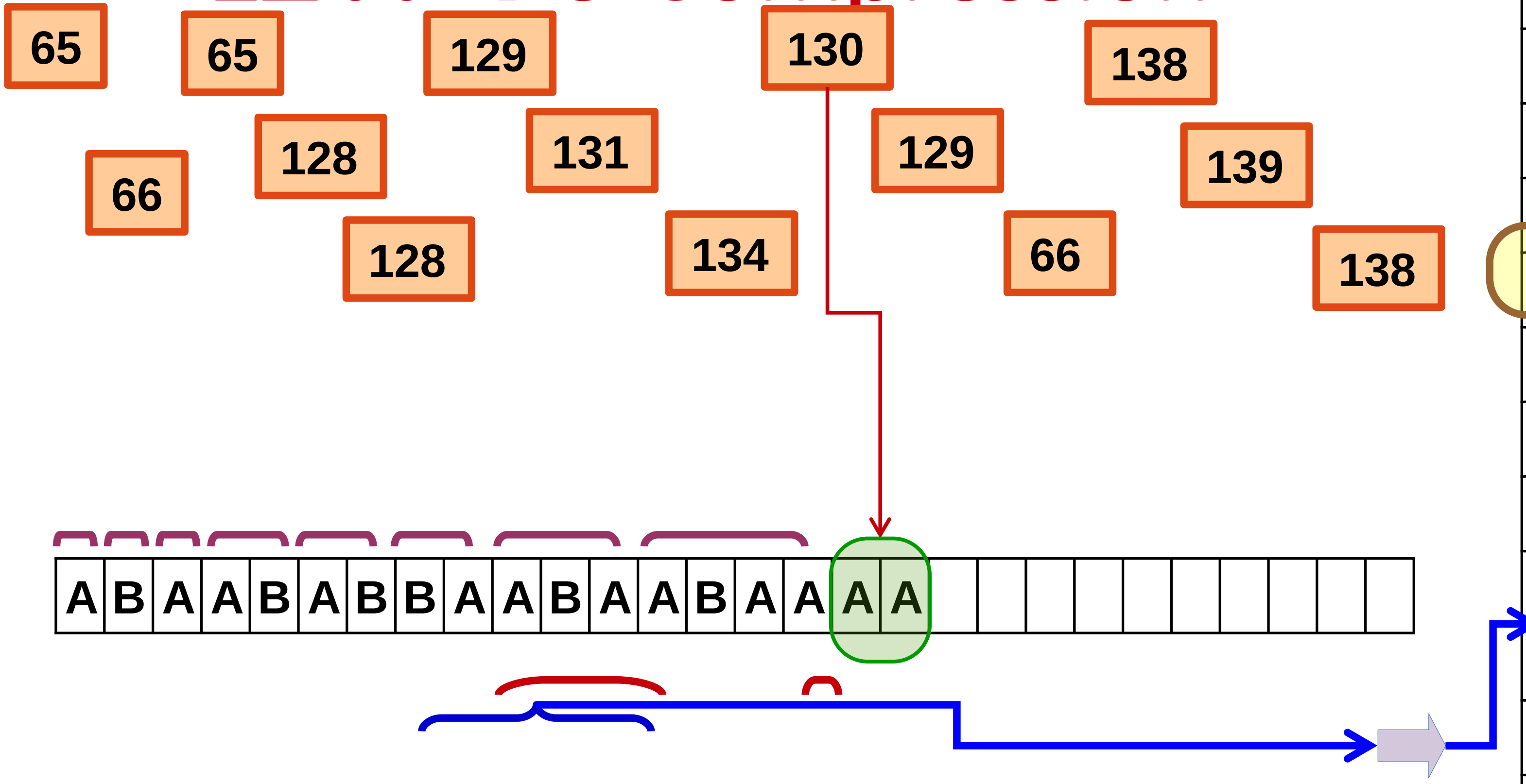


...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	ABAA
135	
136	
137	
138	
139	
140	
141	
142	

Pick symbols at Index [134] from the Dictionary; "ABAA"

Construct Unknown Symbols by Concatenating
Symbols in Previous Step "ABA"
and First Symbol in Previous Step "A"

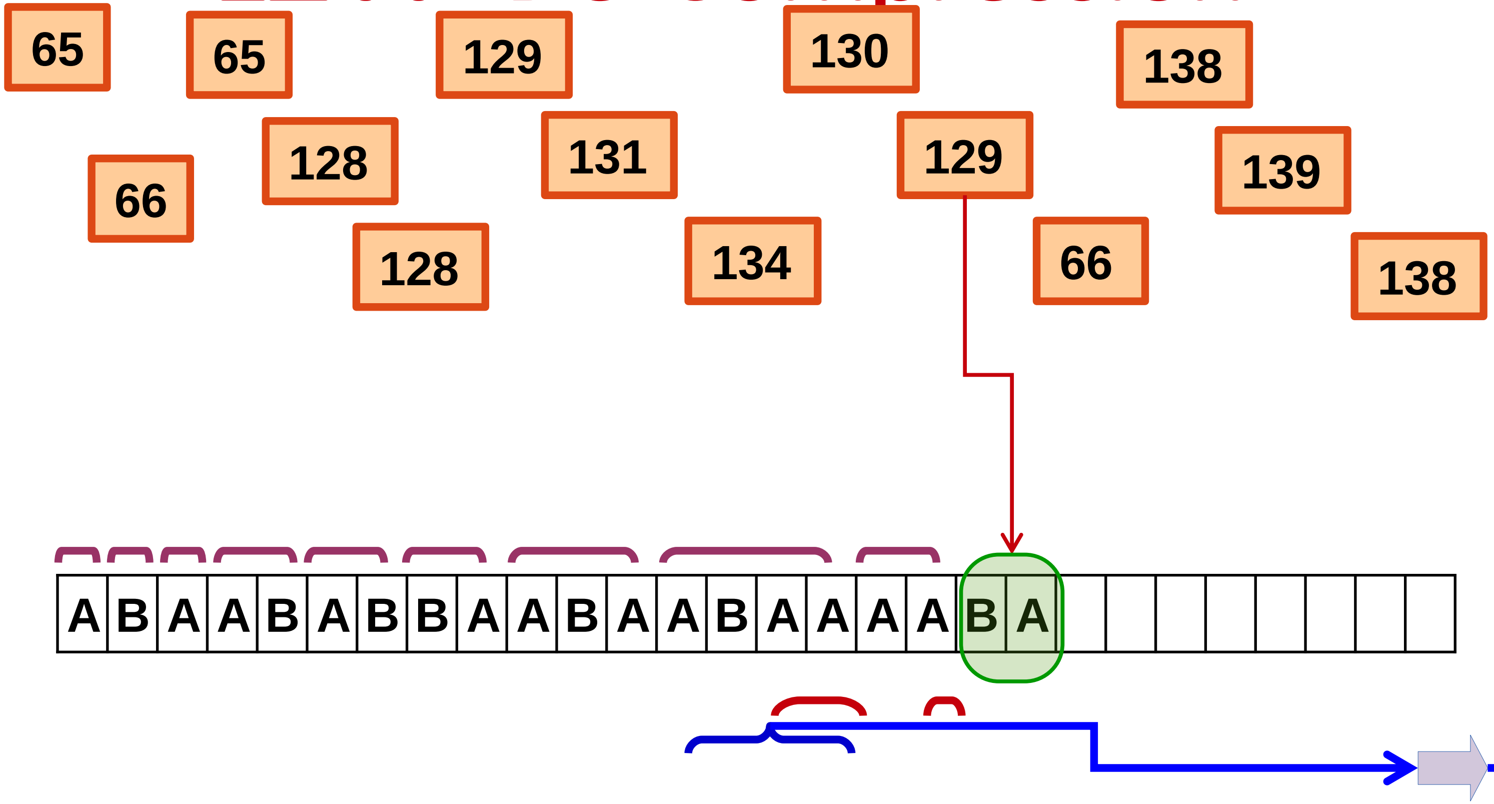
LZW De-Compression



...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	ABAA
135	ABAAA
136	
137	
138	
139	
140	
141	
142	

Pick symbols at Index [130] from the Dictionary; "AA"
Concatenate ***ALL*** *Symbols picked from* ***Previous step***
and ***first*** *Symbol picked from* ***current step***
(Add concatenated Symbols to Dictionary)

LZW De-Compression

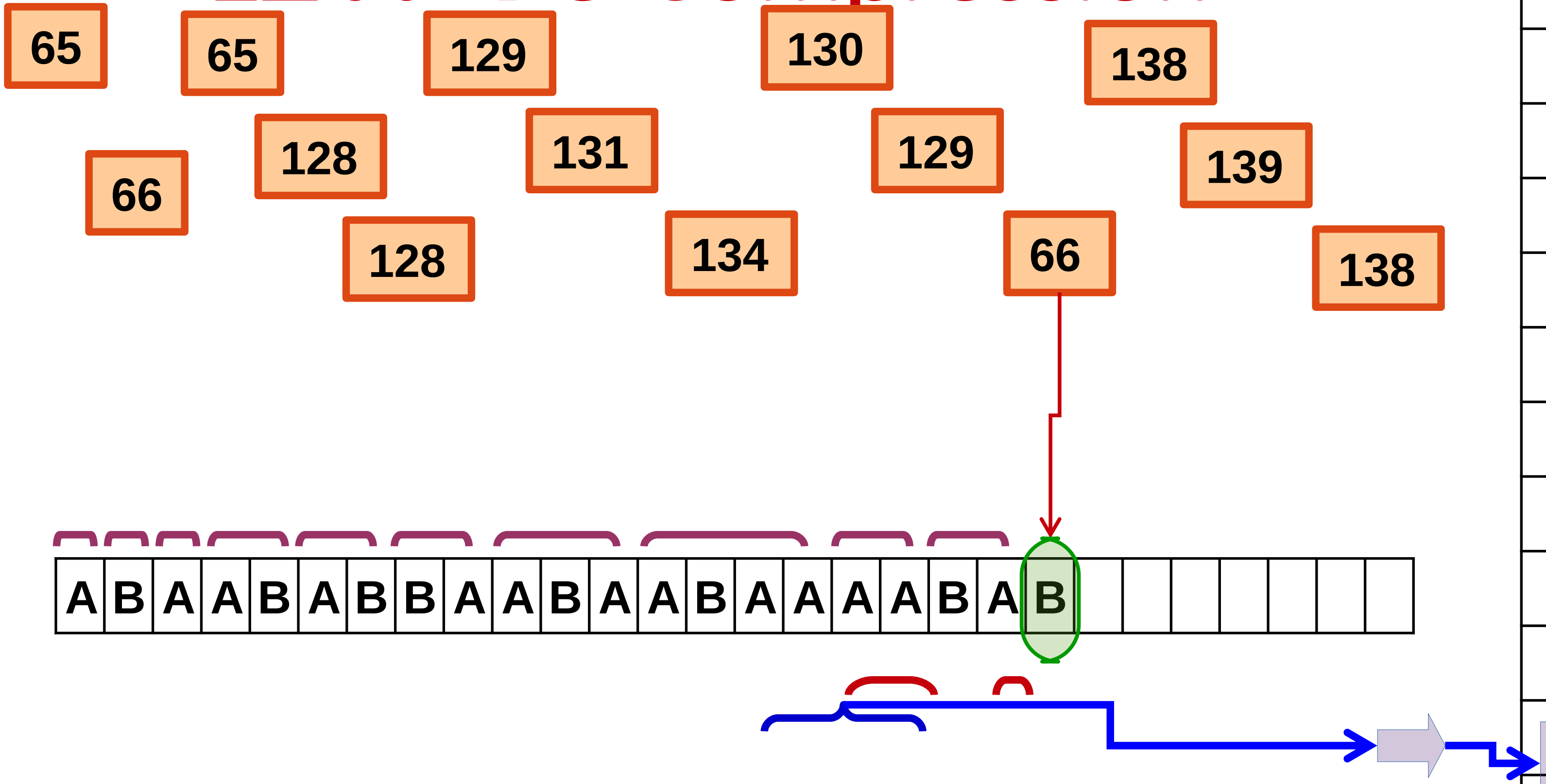


A	B	A	A	B	A	B	B	A	A	B	A	A	B	A	A	A	A	B	A								

...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	ABAA
135	ABAAA
136	AAB
137	
138	
139	
140	
141	
142	

Pick symbols at Index [129] from the Dictionary; "BA"
Concatenate ALL Symbols picked from Previous step
and first Symbol picked from current step
(Add concatenated Symbols to Dictionary)

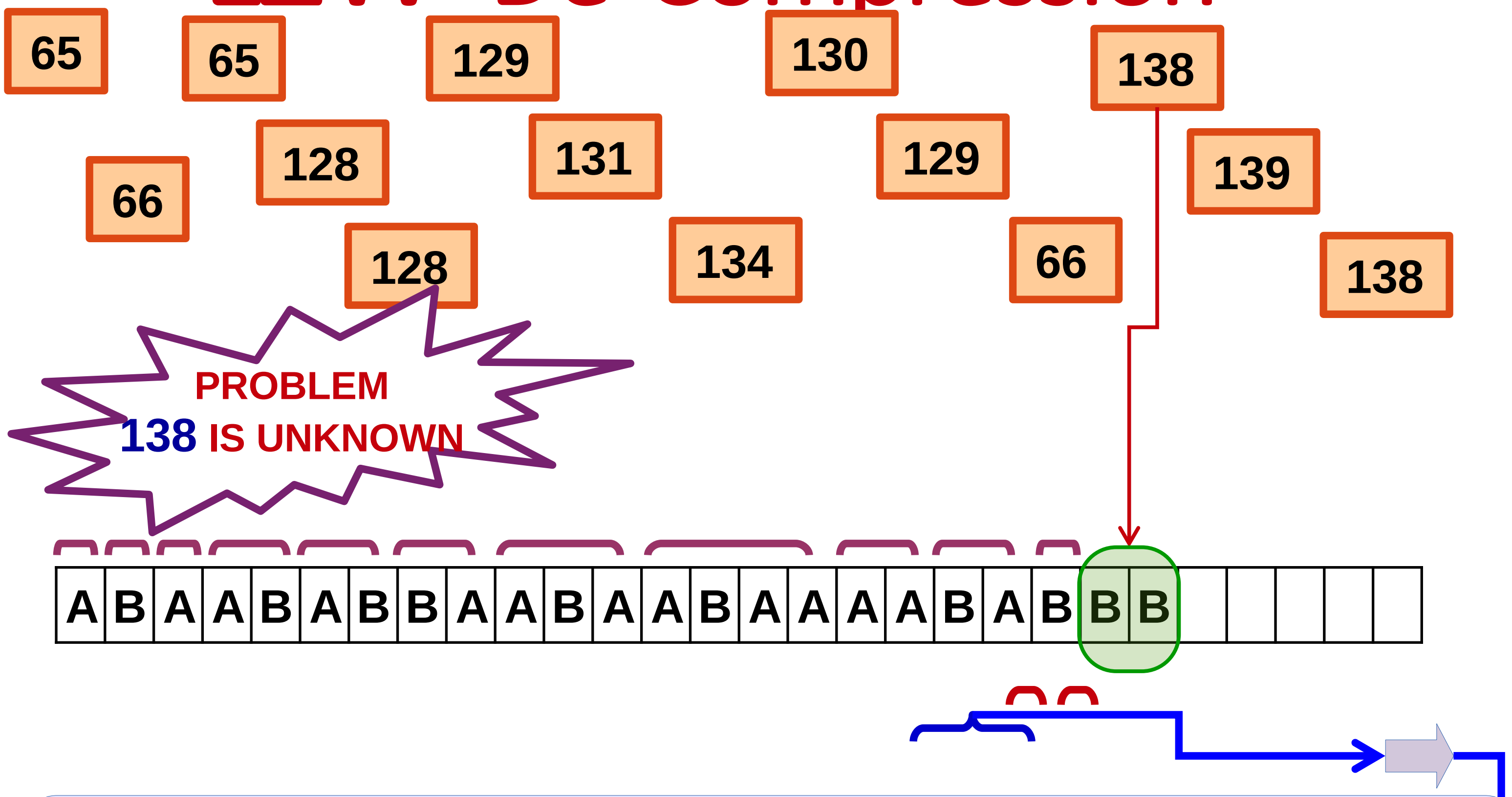
LZW De-Compression



...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	ABAA
135	ABAAA
136	AAB
137	BAB
138	
139	
140	
141	
142	

Pick symbols at Index [98] from the Dictionary; "B"
Concatenate <u>ALL</u> Symbols picked from <u>Previous step</u>
and <u>first</u> Symbol picked from <u>current step</u>
(Add concatenated <u>Symbols</u> to Dictionary)

LZW De-Compression

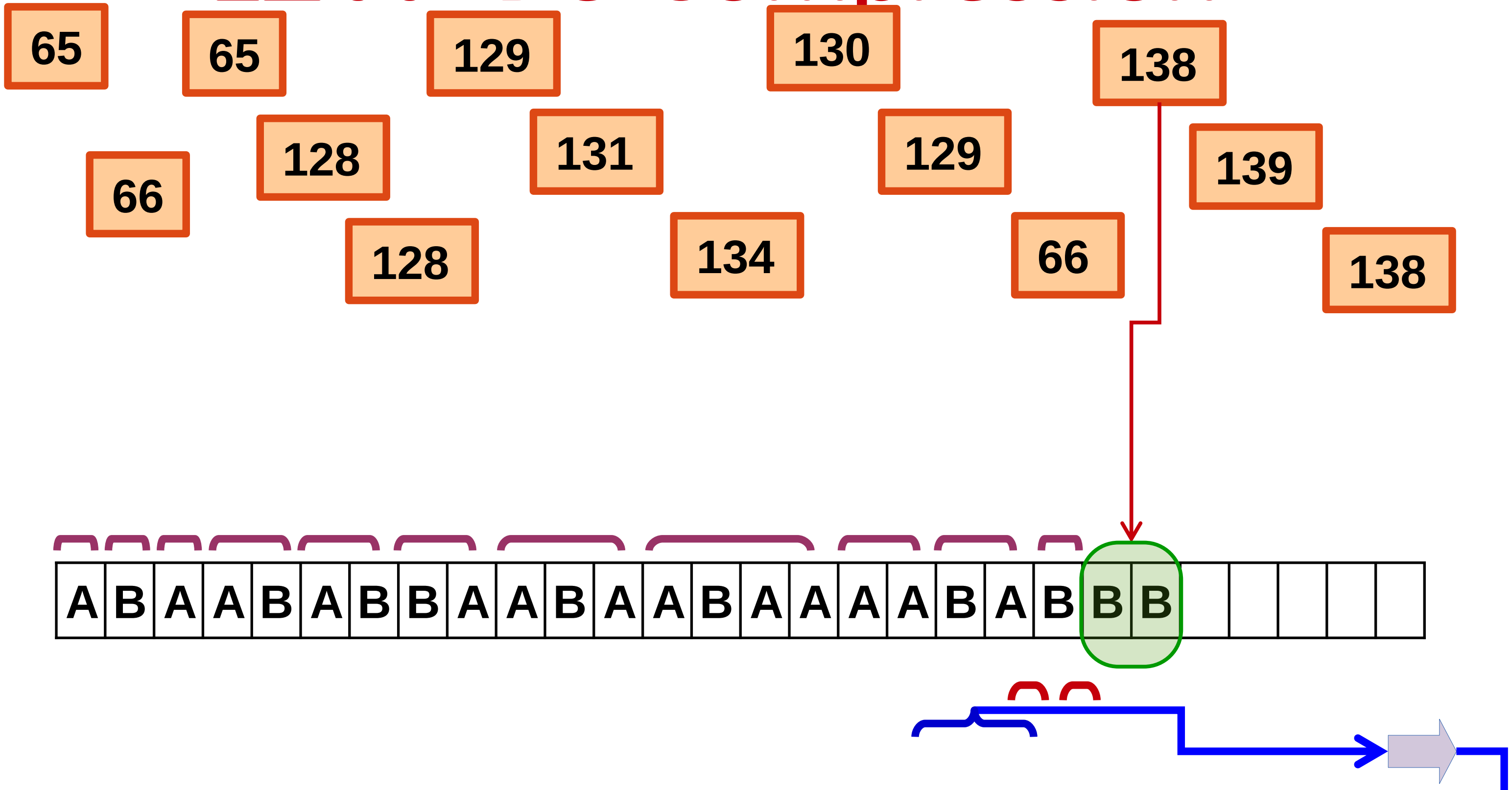


...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	ABAA
135	ABAAA
136	AAB
137	BAB
138	???
139	
140	
141	
142	

Pick symbols at Index [138] from the Dictionary; "???"

Construct Unknown Symbols by Concatenating
Symbols in Previous Step "B"
and First Symbol in Previous Step "B"

LZW De-Compression

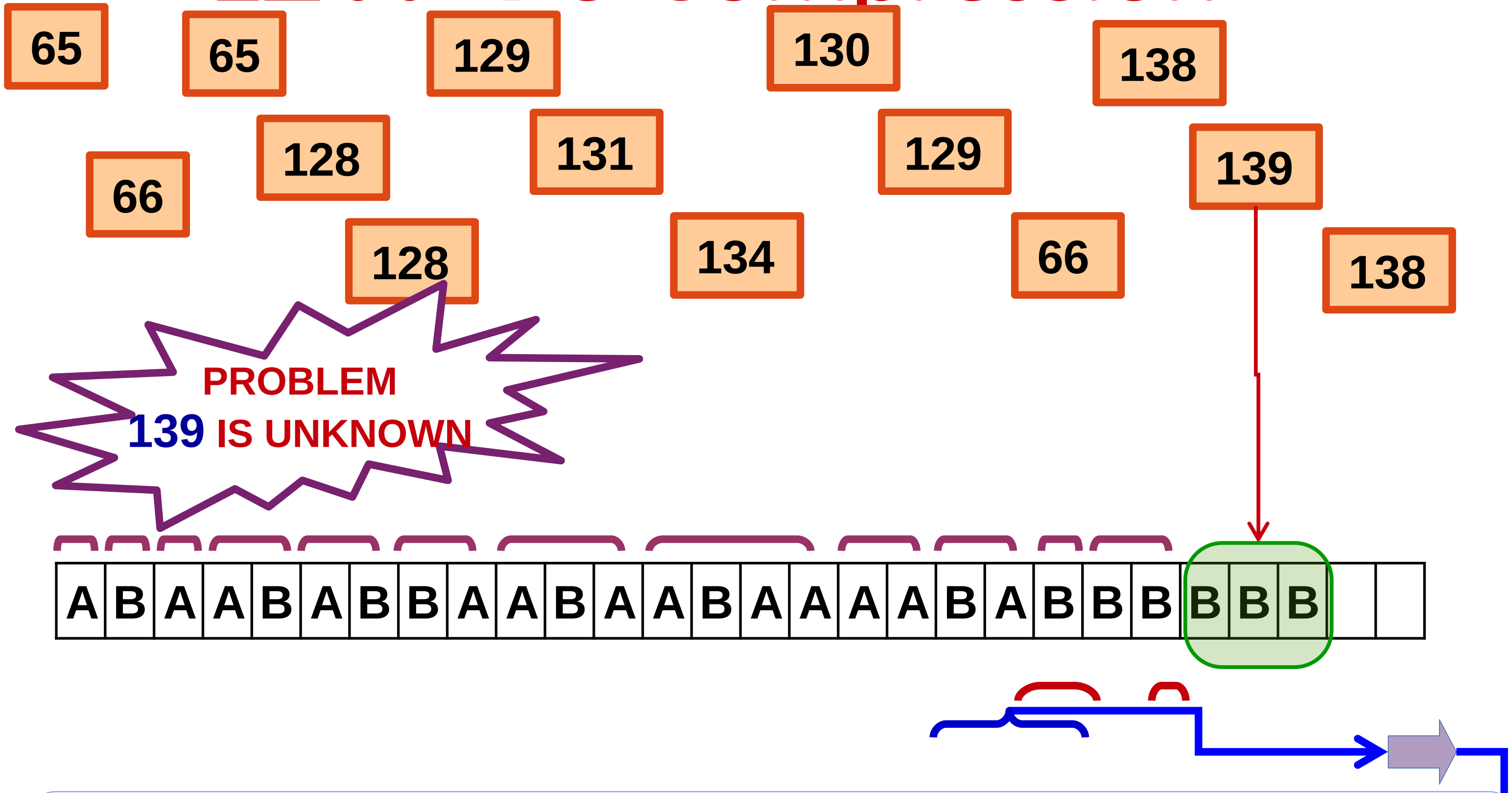


A B A A B A B B A A B A A B A A A A B A B B B

…	…
65	A
66	B
…	…
…	…
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	ABAA
135	ABAAA
136	AAB
137	BAB
138	BB
139	
140	
141	
142	

Concatenate ***ALL Symbols picked from Previous step***
and first Symbol picked from current step
(Add concatenated Symbols to Dictionary)

LZW De-Compression



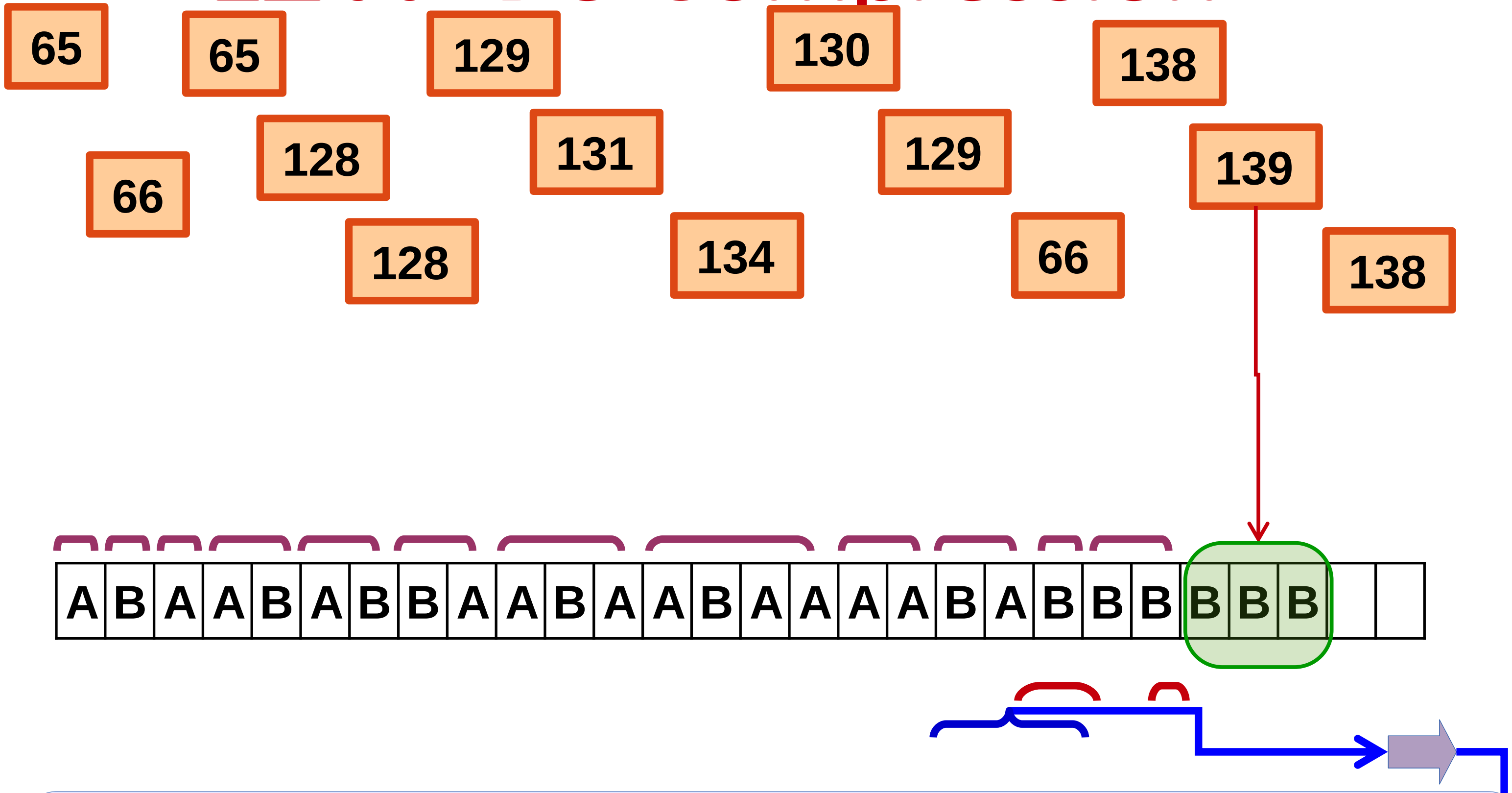
A	B	A	A	B	A	B	B	A	A	B	A	A	B	A	A	A	A	B	A	B	B	B	B	B	B		

…	…
65	A
66	B
…	…
…	…
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	ABAA
135	ABAAA
136	AAB
137	BAB
138	BB
139	???
140	
141	
142	

Pick symbols at Index [139] from the Dictionary; "???"

Construct Unknown Symbols by Concatenating
Symbols in Previous Step "BB"
and First Symbol in Previous Step "B"

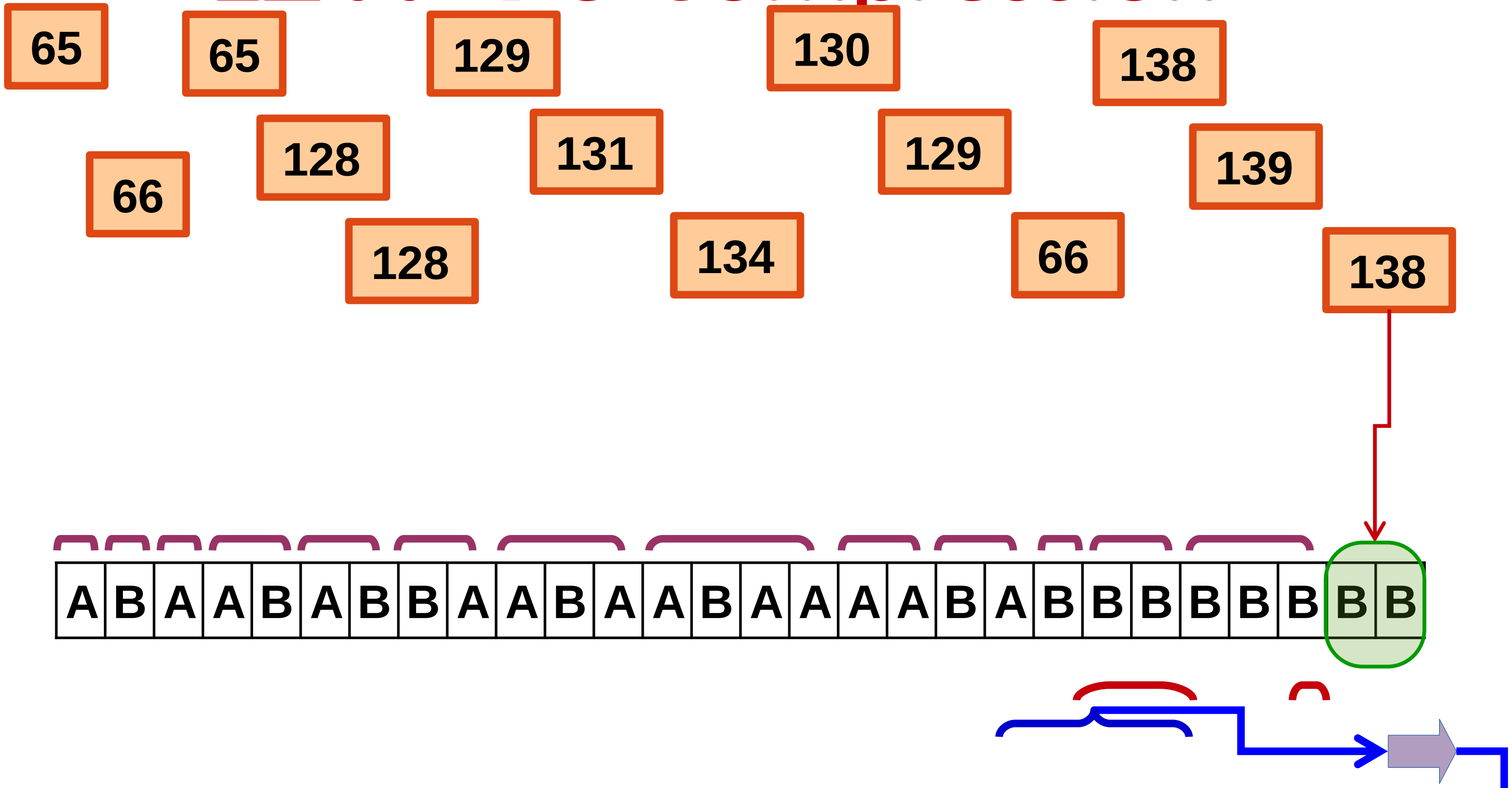
LZW De-Compression



...	...
65	A
66	B
...	...
...	...
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	ABAA
135	ABAAA
136	AAB
137	BAB
138	BB
139	BBB
140	
141	
142	

Concatenate ALL Symbols picked from Previous step and first Symbol picked from current step (Add concatenated Symbols to Dictionary)

LZW De-Compression

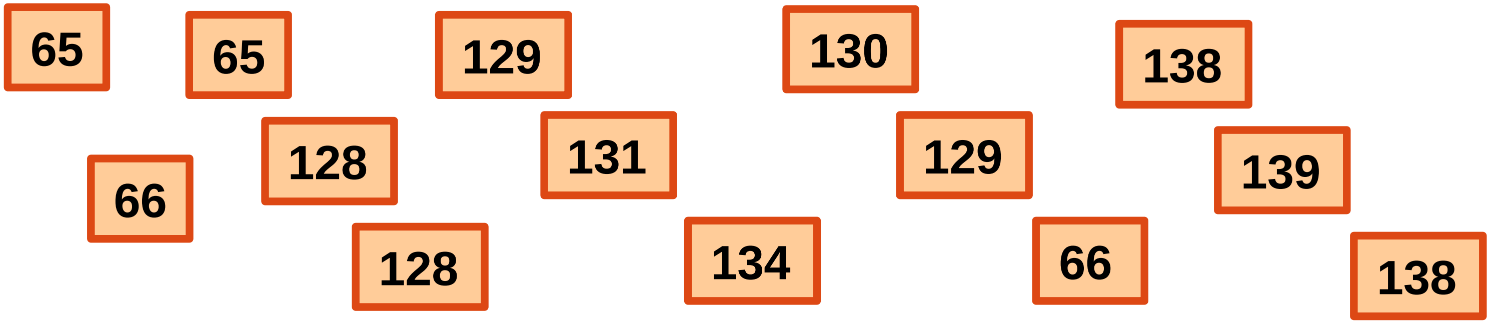


A	B	A	A	B	A	B	B	A	A	B	A	A	B	A	A	A	A	B	A	B	B	B	B	B	B	B	B

…	…
65	A
66	B
…	…
…	…
128	AB
129	BA
130	AA
131	ABA
132	ABB
133	BAA
134	ABAA
135	ABAAA
136	AAB
137	BAB
138	BB
139	BBB
140	BBBB
141	
142	

Pick symbols at Index [138] from the Dictionary; "BB"
Concatenate <u>ALL</u> Symbols picked from <u>Previous step</u>
and <u>first</u> Symbol picked from <u>current step</u>
(Add concatenated <u>Symbols</u> to Dictionary)

LZW Compression Ratio



Original Size = Number of Symbols * Bits used to Store one Symbol
= **28** Symbols * **8** Bits / Symbol = **224** bits
(Store "Symbol" ASCII Code in 8 Bits)

Max "Index" Value = 139 **Store "Index" Value in 8 Bits**

Tag size = 8 Bits

Number of Tags = 14 Tags

Compressed Size=14*8=112 bits

LZW Compression Technique

Advantages:

- Extremely effective when there are repeated patterns in the data that are widely spread
- Prior knowledge of probability of occurrence of symbols to be encoded is not required. Simple coding technique with high compression ratio.
  - LZW compression is fast Lossless compression technique

Disadvantages:

- Creates entries in the dictionary that may never be used.
- LZW is a fairly old compression technique

Dictionary based Techniques LZ 77, LZ 78, LZW

LZ 77 (Triple Tag, No dictionary, sliding window) <2,3,'A'>

LZ 78 (Pair Tag, No window, Dictionary table) <3,'A'>

LZW (single Tag "index", No Window, Dictionary Table)   <130>