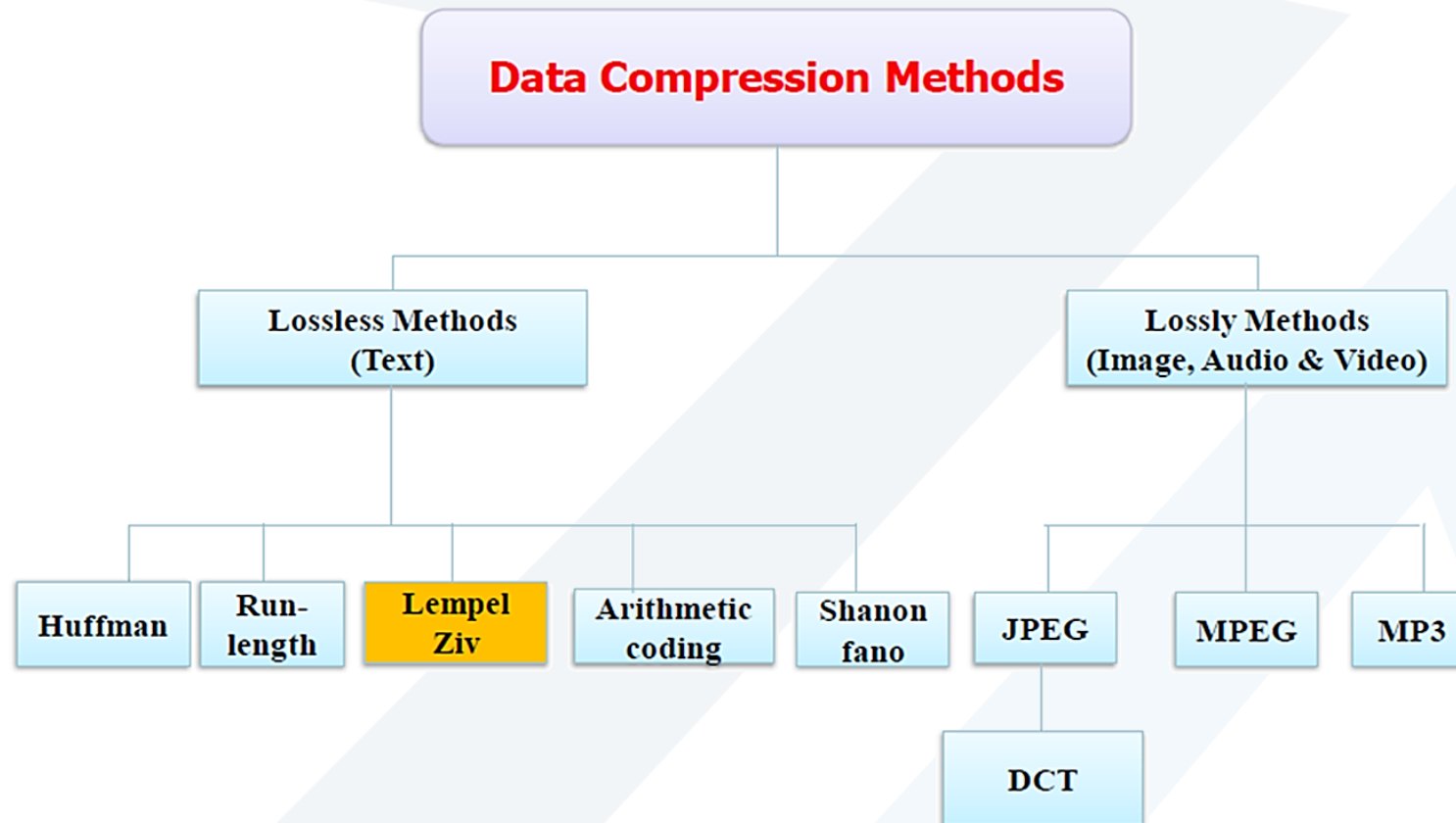


# وسائط متعددة المحاضرة السادسة

أ.د. فادي غصنه

# Lempel-Ziv Code



طرق ضغط البيانات

## Lempel-Ziv Algorithm

-- خوارزمية الضغط LZ تستخدم قطار من الأحرف في حال ترميز النصوص أو القيم المتسلسلة في حال ترميز الصور بدلا من استخدام حرف واحد أو قيمة معينة بواسطة ترميز الـ ASCII. مثال: من أجل ضغط نص ما يتم تشكيل جدول أو قاموس يضم كل احتمالات تسلسل الأحرف وتكرارها في النص من أجل أن ترسل وتعالج بين مرمر وفك الترميز .

- تعرف هذه الخوارزمية بـ Dictionary Coding- (خوارزميات الضغط عن طريق الإستبدال النصي أو الضغط عن طريق القواميس )
- هي خوارزمية ضغط دون ضياع، وتعد أحد أصناف الترميز الإحصائية
- تعتمد على مراقبة العلاقة بين أجزاء البيانات الواردة واستبدال جميع الكلمات المكررة بدليل يشير إليها وهو أقصر منها وذلك في جدول dictionary الذي يحوي الكلمات الواردة في النص المراد ترميزه.

## Lempel-Ziv Algorithm

-- بدلا من إرسال كلمة ترميز بـ **ASCII-codewords** لكل حرف في النص أو قيمة في مصفوفة الصورة المراد ترميزها، يتم إرسال عنوان موقع القيمة أو الحرف المراد ترميزه في القاموس المشكل أي مهما كان طول الحرف يتم ترميزه بعدد ثابت من البتات يدل على موقع الحرف أو القيمة في القاموس المخزن في كل من المرسل والمستقبل.

-- في المستقبل يقوم فك الترميز باستخدام العنوان المرسل للموقع حتى يصل إلى الكلمة أو القيمة المناسبة من القاموس أو الجدول المخزن لديه حتى يبدأ عملية بناء النص أو الصورة.

➤ تصنف خوارزميات الضغط (LZ) إلى صنفين:

i. LZ77

ii. LZ78

تتفق الخوارزميتين من حيث المبدأ.

## الخوارزمية LZ78

### مبدأ العمل

تعتمد هذه الخوارزمية على بناء جدول يتضمن كل الكلمات الواردة في النص لمرة واحدة (أول مرة)، بحيث لا يكون هناك حاجة عند الترميز لإرسال الكلمة عند ورودها في النص مرة أخرى وإنما يرسل دليلاً يشير إلى موقع هذه الكلمة ضمن الجدول.

➤ عملية بناء القاموس dictionary

➤ في حال ورود الكلمة لأول مرة:

ترمز بكلمة ترميز وتكون مؤلفة من جزأين هما الدليل والكلمة نفسها : (index,word)

➤ في حال كانت الكلمة في الجدول:

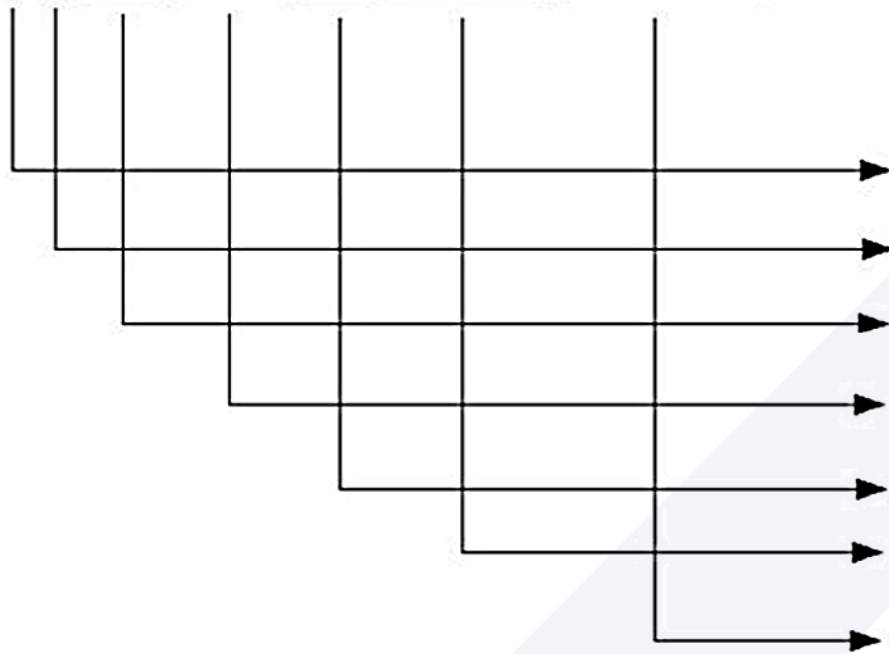
هنا لا داعي لإرسال الكلمة ويتم إرسال الدليل فقط كالتالي (index,)

(0, char)	if one-character pattern is not in Dictionary.
(DictionaryPrefixIndex, lastPatternCharacter)	if multi-character pattern is not in Dictionary.
(DictionaryPrefixIndex, )	if the last input character or the last pattern is in the Dictionary.

يعرف هذا الجدول بالقاموس المتكيف ( adaptive dictionary ) :  
 أي أنه لا حاجة لأن يكون فاك الترميز على علم مسبق بمحتويات هذا الجدول، وإنما يقوم  
 ببناء جدولته حالما يستقبل البيانات الواردة وذلك بألية مشابهة للآلية التي يعمل بها المرمز

## مثال 1 عن الخوارزمية LZ78

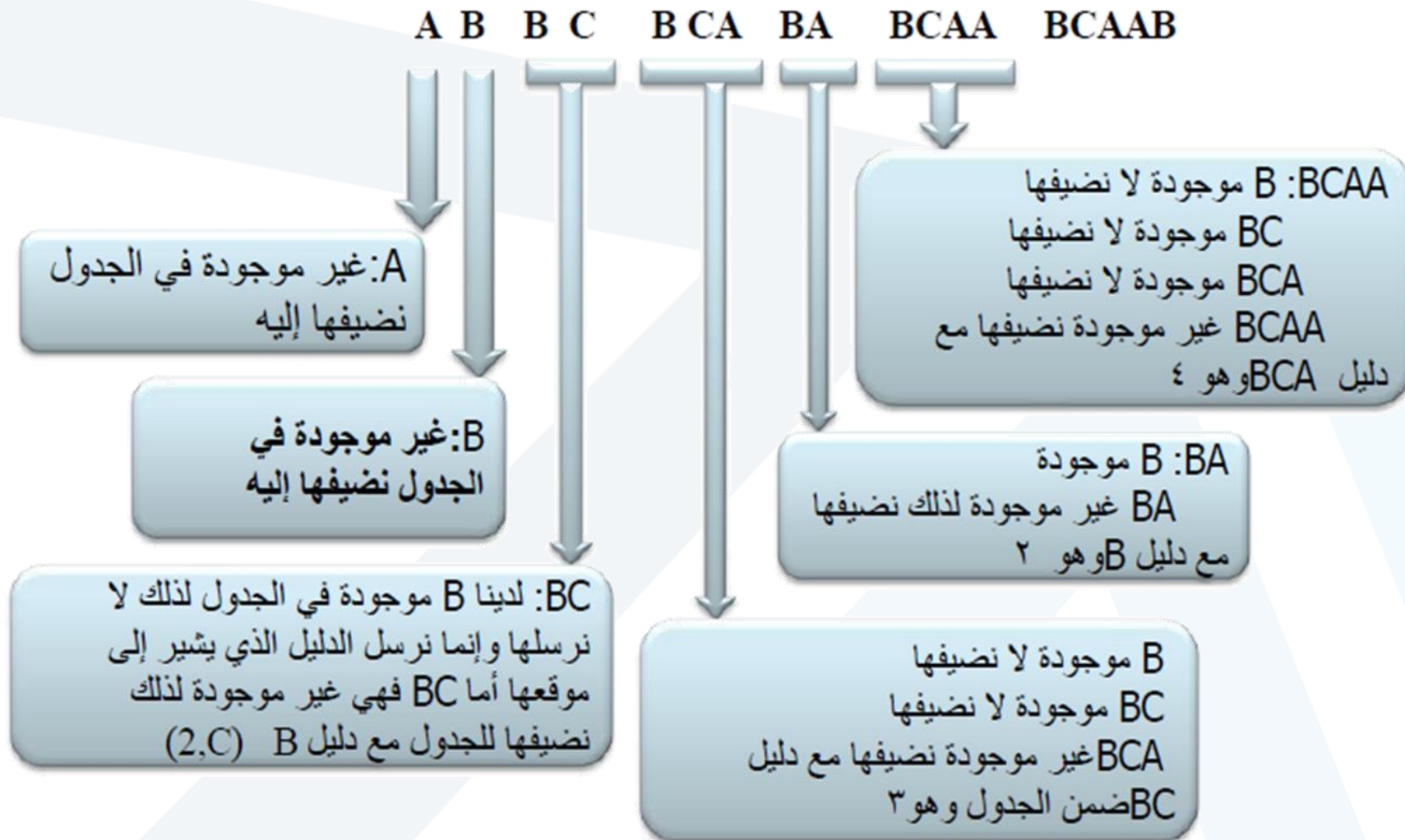
1 2 3 4 5 6 7  
A B BC BCA BA BCAA BCAAB



Dictionary		
output	index	string
(0, A)	1	A
(0, B)	2	B
(2, C)	3	BC
(3, A)	4	BCA
(2, A)	5	BA
(4, A)	6	BCAA
(6, B)	7	BCAAB

المطلوب ترميز سلسلة المحارف الآتية:

ABBCBCABABCAABCAAB





- يرمز الجزء الصحيح من كلمة الترميز بعدد بتات تمثل الترميز الثنائي لـ (الدليل -1)
- مثلاً (4,A) حسب الرسالة المرمزة دليل كلمة الترميز هذه هو 6 و الجزء الصحيح هو 4:
- 4 ترمز ب 100 ➔ لترميز العدد 5 إلى 3 خانات ➔ Index-1=5 ➔ Index=6
- و منه:

الرسالة المرمزة: (0,A)(0,B)(2,C)(3,A)(2,A)(4,A)(6,B)

index

1 2 3 4 5 6 7

0A0B10C11A010A100A110B

➤ مقدار الضغط = عدد البتات المطلوبة لترميز هذه الرسالة بالأسكي مقسوماً على عدد البتات اللازمة للترميز باستخدام الخوارزمية LZ78

➤ يكون عدد البتات المطلوبة لترميز هذه الرسالة بلغة ASCII هو:

$$7 * 18 = 126 \text{ bits}$$

➤ يكون عدد البتات المطلوبة للترميز باستخدام الخوارزمية LZ78 :

0A	0B	10C	11A	010A	100A	110B
(1+7)	(1+7)	(2+7)	(2+7)	(3+7)	(3+7)	(3+7)

$$(1+7) + (1+7) + (2+7) + (2+7) + (3+7) + (3+7) + (3+7) = 64 \text{ BITS}$$

$$\text{مقدار الضغط: } 126 / 64 = 1.96$$

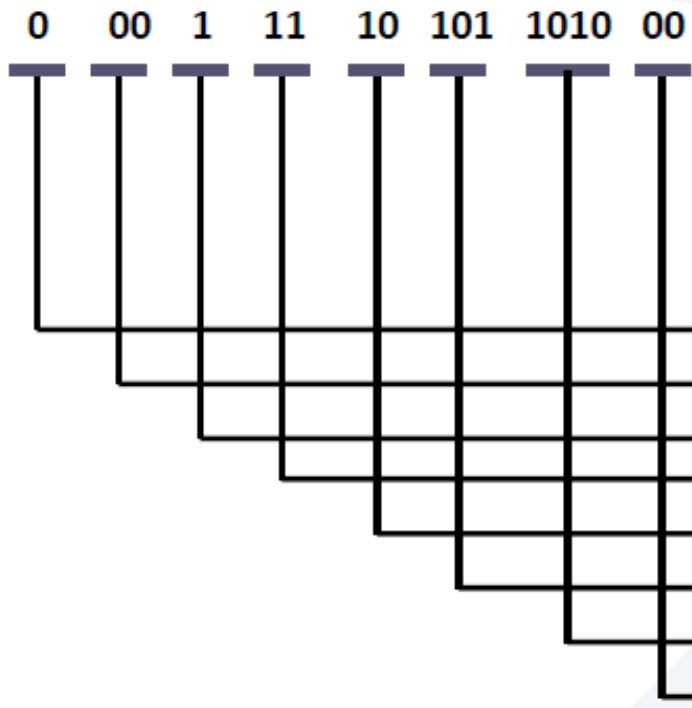
index	index-1	bits	Number of significant bits
1	0	0	1
2	1	1	
3	2	10	2
4	3	11	
5	4	100	3
6	5	101	
7	6	110	
8	7	111	
9	8	1000	4
10	9	1001	
11	10	1010	
12	11	1011	
13	12	1100	
14	13	1101	
15	14	1110	
16	15	1111	

Codeword (0, A) (0, B) (2, C) (3, A) (2, A) (4, A) (6, B)

index 1 2 3 4 5 6 7

Bits:  $(1 + 7) + (1 + 7) + (2 + 7) + (2 + 7) + (3 + 7) + (3 + 7) + (3 + 7) = 64$  bits

## مثال 2 عن الخوارزمية LZ78



المطلوب ترميز السلسلة الآتية وحساب مقدار الضغط:

00011110101101000

Output	Dictionary	
	Index	string
(0, 0)	1	0
(1, 0)	2	00
(0, 1)	3	1
(3, 1)	4	11
(3, 0)	5	10
(5, 1)	6	101
(6, 0)	7	1010
(2, )	8	00

الرسالة المرمزة = (0,0) (1,0) (0,1) (3,1) (3,0) (5,1)(6,0) (2, )

## حساب مقدار الضغط:

يكون عدد البتات المطلوبة للترميز باستخدام الخوارزمية LZ78 :

	(0,0)	(1,0)	(0,1)	(3,1)	(3,0)	(5,1)	(6,0)	(2, )
<b>index</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
	<b>00</b>	<b>10</b>	<b>001</b>	<b>111</b>	<b>0110</b>	<b>1011</b>	<b>1100</b>	<b>010</b>
	(1+7)	(1+7)	(2+7)	(2+7)	(3+7)	(3+7)	(3+7)	3
	<b>(1+7)+ (1+7)+ (2+7)+ (2+7)+ (3+7)+(3+7)+(3+7)+3=67 bits</b>							

index	index-1	bits	Number of significant bits
1	0	0	1
2	1	1	
3	2	10	2
4	3	11	
5	4	100	3
6	5	101	
7	6	110	
8	7	111	
9	8	1000	4
10	9	1001	
11	10	1010	
12	11	1011	
13	12	1100	
14	13	1101	
15	14	1110	
16	15	1111	

Codeword (0,0) (1,0) (0,1) (3,1) (3,0) (5,1) (6,0) (2, )

index 1 2 3 4 5 6 7 8

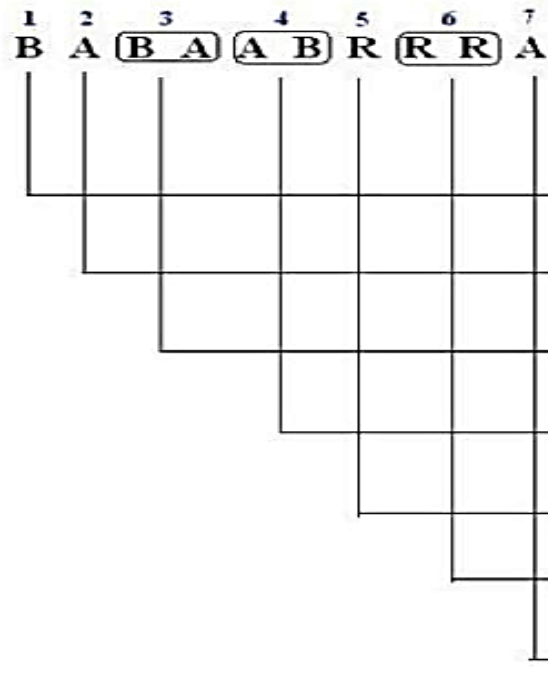
Bits:  $(1 + 7) + (1 + 7) + (2 + 7) + (2 + 7) + (3 + 7) + (3 + 7) + (3 + 7) + (3) =$   
67 bits

➤ يكون عدد البتات المطلوبة لترميز هذه الرسالة بلغة ASCII هو:

$$7 * 17 = 119 \text{ bits}$$

➤ يكون مقدار الضغط :  $119/67 = 1.77$

## مثال 3 عن الخوارزمية LZ78



Dictionary		
output	index	string
(0, B)	1	B
(0, A)	2	A
(1, A)	3	BA
(2, B)	4	AB
(0, R)	5	R
(5, R)	6	RR
(2, )		

المطلوب ترميز سلسلة المحارف الآتية:

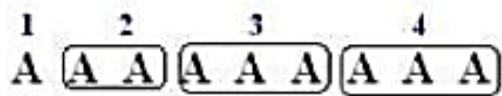
**BABAABRRRA**

وهكذا تكون الرسالة المرمزة: (0, B)(0, A)(1, A)(2, B)(0, R)(5, R)(2, )



## مثال 4 عن الخوارزمية LZ78

المطلوب ترميز سلسلة المحارف الآتية: AAAAAAAAAA

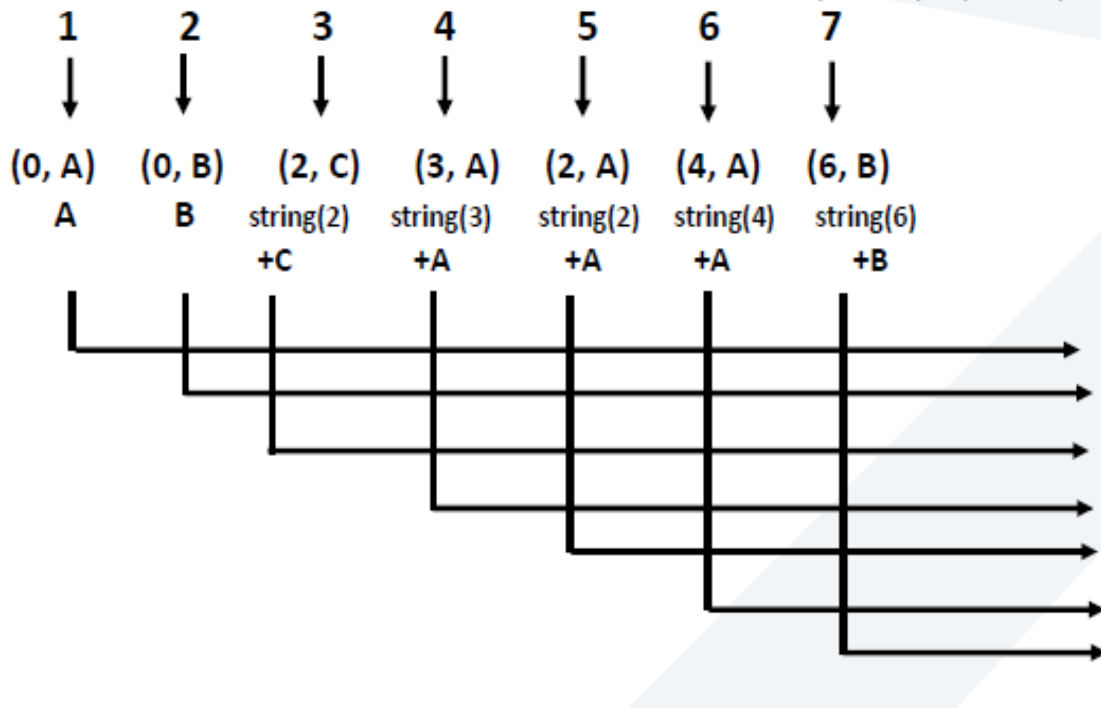


Dictionary		
output	Index	string
(0, A)	1	A
(1, A)	2	AA
(2, A)	3	AAA
(3, )		

وهكذا تكون الرسالة المرمزة: (0, A)(1, A)(2, A)(3, )

## مثال 5 عن الخوارزمية LZ78

المطلوب: فك ترميز الرسالة : (0, A) (0, B) (2, C) (3, A) (2, A) (4, A) (6, B)



Dictionary		
Output	Index	string
A	1	A
B	2	B
BC	3	BC
BCA	4	BCA
BA	5	BA
BCAA	6	BCAA
BCAAB	7	BCAAB

الرسالة بعد فك ترميزها: **ABBCBCABABCAABCAAB**

## مثال 6 عن الخوارزمية LZ78

المطلوب: فك ترميز الرسالة الآتية المرمنة باستخدام LZ78:

(0, w) (0, a) (0, b) (3, a) (0, d) (1, a) (3, b) (2, d) (6, b) (4, d) (9, b)  
(8, w) (0, o) (13, d) (1, o) (14, w) (13, o) (6,)

الحل:

1	2	3	4	5	6	7	8	9	10	11
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
(0, w)	(0, a)	(0, b)	(3, a)	(0, d)	(1, a)	(3, b)	(2, d)	(6, b)	(4, d)	(9, b)
w	a	b	string(3)	d	string(1)	string(3)	string(2)	string(6)	string(4)	string(9)
			+a		+a	+b	+d	+b	+d	+b
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
w	a	b	ba	d	wa	bb	ad	wab	bad	wabb ...

وهكذا نتابع الحل فتكون الرسالة بعد فك ترميزها:

wabbadwabbadwabbadwabbadwoodwoodwoowa

## مثال 7 عن الخوارزمية LZ78

المطلوب فك ترميز الرسالة الآتية: (0, B)(0, A)(1, A)(2, B)(0, R)(5, R)(2, )

Dictionary		
output	index	string
B	1	B
A	2	A
BA	3	BA
AB	4	AB
R	5	R
RR	6	RR
A		

وهكذا تكون الرسالة بعد فك ترميزها: BABAABRRRA

## مثال 8 عن الخوارزمية LZ78

المطلوب فك ترميز الرسالة الآتية: (0, A)(1, A)(2, A)(3, )

Dictionary		
output	index	string
A	1	A
AA	2	AA
AAA	3	AAA
AAA		

وهكذا تكون الرسالة بعد فك ترميزها: AAAAAAAAAA

## إيجابيات وسلبيات الخوارزمية LZ

### إيجابيات الخوارزمية `lempel zif`

- ❖ تحقق نسبة ضغط كبيرة وفعالة من أجل النصوص الكبيرة
- ❖ لا داعي لأن يحتوي فاك الترميز على (dictionary) مسبقاً، لأنه يقوم ببنائه اعتماداً على البيانات الواردة بالتتالي

### سلبيات الخوارزمية `lempel zif`

- ❖ غير فعالة في ضغط النصوص الصغيرة

## خوارزمية LZW

- تم تطوير خوارزمية Lempel–Ziv فيما بعد من قبل Terry A. Welch
- تم تسمية هذه الخوارزمية بخوارزمية LZW.
- يعتمد مبدأ هذه الخوارزمية على بناء محتويات القاموس ديناميكياً حالما يتم إرسال النص في كلا الطرفين المرمز وفاك الترميز
- تتضمن المداخل الـ 128 الأولى في القاموس المحارف المفردة التي تشكل مجموعة الأسكي.
- تحتوي بقية محتويات ومداخل القاموس محرفين أو أكثر (كلمات).
- سيعلم القاموس بعد كل space أنها كلمة جديدة.

## مثال عن خوارزمية LZW

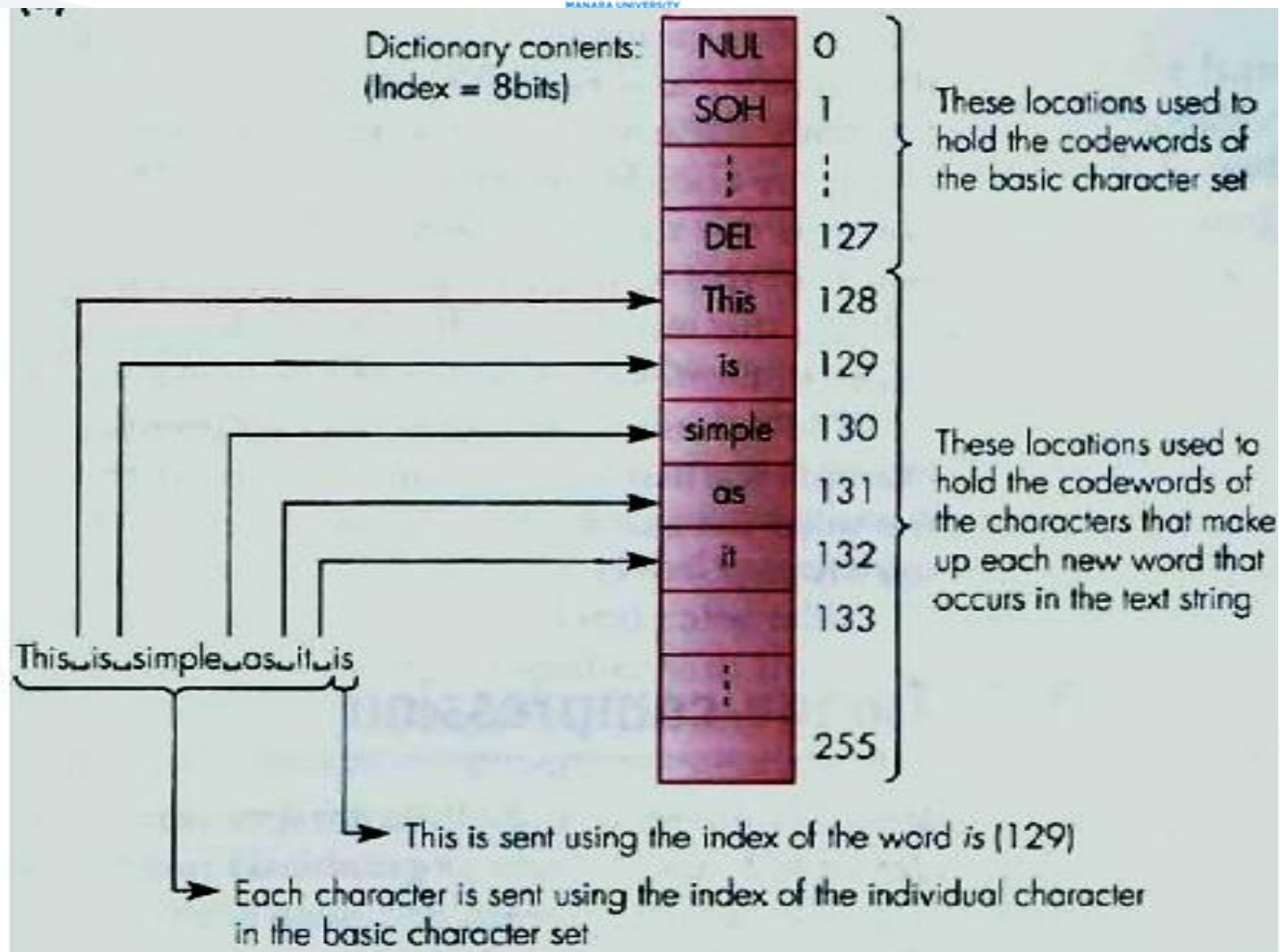
➤ نستعرض آلية تمثيل العبارة (This is simple as it is) في القاموس.

✓ المواقع من (0) وحتى (127): تستخدم لتحميل كلمات الترميز لمجموعة الحروف الأساسية.

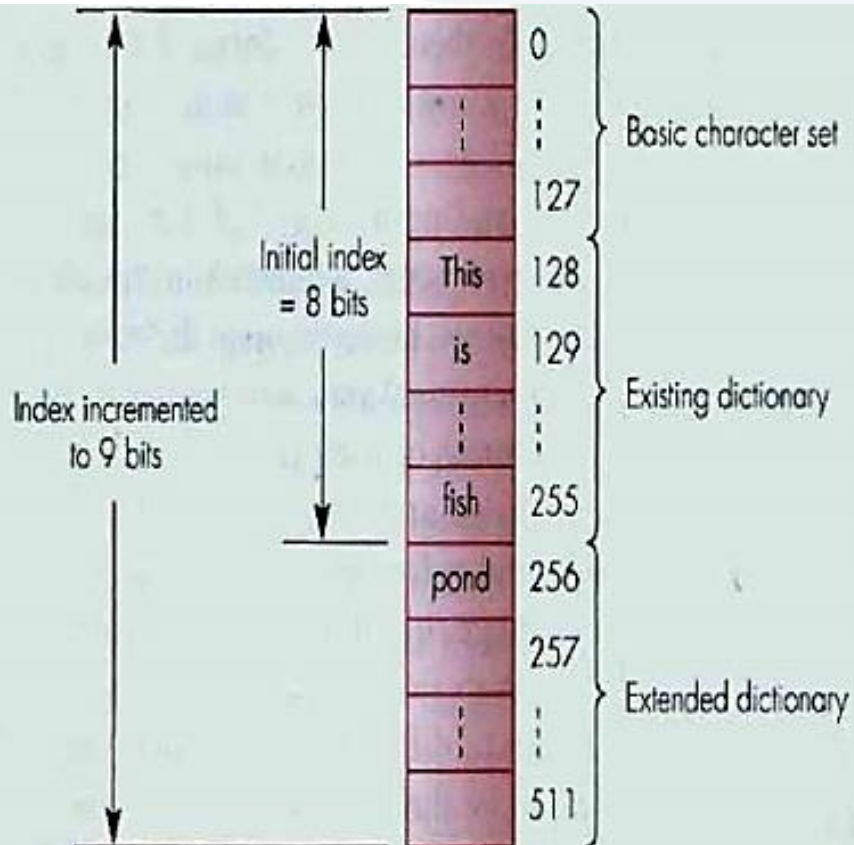
✓ المواقع من (128) وحتى (255): تستخدم لتحميل كلمات ترميز الحروف التي تتركب كل كلمة جديدة تحصل في سلسلة حروف النص.

✓ نلاحظ من الشكل أن الكلمة (is) مكررة وكان دليلها هو (129)، وبالتالي عند ورودها في النص مرة أخرى يتم ارسال دليل الكلمة فقط.





## خوارزمية LZW



-- في حال لدينا تطبيق ما مكون من 128 حرف أو قيمة أساسية،  
إذا القاموس في كل من المرمز وفاق الترميز سيبدأ بـ  
مدخل  $256=128(\text{ASCII})+128(\text{word or values})=2^8$

-- من اجل تمديد القاموس ديناميكيًا من اجل ملائمة زيادة المداخل  
لعدد جديد نضيف بت واحد لترميز المواقع وفق  $2^9=512$  مدخل.

-- في المثال المبين جانبًا نلاحظ أن آخر دخل في القاموس الموجود  
هي كلمة *Fish* الكلمة التالية هي *Pond* وغير موجودة في  
القاموس الأساسي، لذلك يتم تمديد القاموس بمقدار 1 Bit للعناوين  
مما يزيد الأمكنة إلى  $2^9=512$ .