Name: ROTHA Dapravith

ID: e20190915

Group: I5-GIC(B)

**Assignment Lesson 09**

Question

1) Find encoder and decoder of LZW? If we have:

Input string: “abdcaedbdcecabb”

2) Find encoder and decoder of LZW? If we have:

Input string: “cdaabbefacbdegfeeggf”

**Answers**

1). Find encoder and decoder of LZW

Given if we have: Input string: “abdcaedbdcecabb”

* Encoder: First, we find all different characters.

There are “a,b,c,d,e”

Then build a base dictionary table:

|  |  |
| --- | --- |
| **Entry** | **Codeword** |
| a | 1 |
| b | 2 |
| c | 3 |
| d | 4 |
| e | 5 |

Next, Build an output dictionary we got:

a b d c a e d b d c e c a b b EOF

* s = 'a', c = 'b',
* since 'ab' is not in the dictionary,
* output codeword for 'a' (1),
* add 'ab' to dictionary with codeword 6.
* s = 'b', c = 'd',
* since 'bd' is not in the dictionary,
* output codeword for 'b' (2),
* add 'bd' to dictionary with codeword 7.
* s = 'd', c = 'c', since 'dc' is not in the dictionary,
* output codeword for 'd' (4),
* add 'dc' to dictionary with codeword 8.
* s = 'c', c = 'a',
* since 'ca' is not in the dictionary,
* output codeword for 'c' (3),
* add 'ca' to dictionary with codeword 9.
* s = 'a', c = 'e',
* since 'ae' is not in the dictionary,
* output codeword for 'a' (1),
* add 'ae' to dictionary with codeword 10.
* s = 'e', c = 'd',
* since 'ed' is not in the dictionary,
* output codeword for 'e' (5),
* add 'ed' to dictionary with codeword 11.
* s = 'd', c = 'b',
* since 'db' is not in the dictionary,
* output codeword for 'd' (4),
* add 'db' to dictionary with codeword 12.
* s = 'b', c = 'd',
* since 'bd' is already in the dictionary,
* set s to 'bd' and move to the next c which is 'c'.
* s = 'bd', c = 'c',
* since 'bdc' is not in the dictionary,
* output codeword for 'bd' (7),
* add 'bdc' to dictionary with codeword 13.
* s = 'c', c = 'e',
* since 'ce' is not in the dictionary,
* output codeword for 'c' (3),
* add 'ce' to dictionary with codeword 14.
* s = 'e', c = 'c',
* since 'ec' is not in the dictionary,
* output codeword for 'e' (5),
* add 'ec' to dictionary with codeword 15.
* s = 'c', c = 'a',
* since 'ca' is already in the dictionary,
* set s to 'ca' and move to the next c which is 'b'.
* s = 'ca', c = 'b',
* since 'cab' is not in the dictionary,
* output codeword for 'ca' (9),
* add 'cab' to dictionary with codeword 16.
* s = 'b', c = 'b', since 'bb' is not in the dictionary,
* output codeword for 'b' (2),
* add 'bb' to dictionary with codeword 17.
* s = 'b', since this is the last character,
* output codeword for 'b' (2), and we reach EOF.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **s** | **c** | **Item** | **Codeword** | **Output** |
| a | b | ab | 6 | 1 |
| b | d | bd | 7 | 2 |
| d | c | dc | 8 | 4 |
| c | a | ca | 9 | 3 |
| a | e | ae | 10 | 1 |
| e | d | ed | 11 | 5 |
| d | b | db | 12 | 4 |
| bd | c | bdc | 13 | 7 |
| c | e | ce | 14 | 3 |
| e | c | ec | 15 | 5 |
| ca | b | cab | 16 | 9 |
| b | b | bb | 17 | 2 |
| b | EOF | - | - | 2 |

Thus, encoder is {1, 2, 4, 3, 1, 5, 4, 7, 3, 5, 9, 2, 2}

* Deconder : Input string is the result of encoder.

String = 1 2 4 3 1 5 4 7 3 5 9 2 2 EOF

* Output dictionary:
* Step 1:

➢ s = null and k = 1.

➢ entry = "a" (codeword 1 → "a").

➢ k is in the base dictionary.

➢ Output entry ("a"). s is empty.

➢ s = entry = "a".

* Step 2:

➢ s = "a" and k = 2.

➢ entry = "b" (codeword 2 → "b").

➢ k is in the base dictionary.

➢ Output entry ("b"). s is not empty.

➢ Add s + entry[0] = "ab". s = entry = "b".

* Step 3:

➢ s = "b" and k = 4.

➢ entry = "d" (codeword 4 → "d").

➢ k is in the base dictionary.

➢ Output entry ("d"). s is not empty.

➢ Add s + entry[0] = "bd". s = entry = "d".

* Step 4:

➢ s = "d" and k = 3.

➢ entry = "c" (codeword 3 → "c").

➢ k is in the base dictionary.

➢ Output entry ("c"). s is not empty.

➢ Add s + entry[0] = "dc". s = entry = "c".

* Step 5:

➢ s = "c" and k = 1.

➢ entry = "a" (codeword 1 → "a").

➢ k is in the base dictionary.

➢ Output entry ("a"). s is not empty.

➢ Add s + entry[0] = "ca". s = entry = "a".

* Step 6:

➢ s = "a" and k = 5.

➢ entry = "e" (codeword 5 → "e").

➢ k is in the base dictionary.

➢ Output entry ("e"). s is not empty.

➢ Add s + entry[0] = "ae". s = entry = "e".

* Step 7:

➢ s = "e" and k = 4.

➢ entry = "d" (codeword 4 → "d").

➢ k is in the base dictionary.

➢ Output entry ("d"). s is not empty.

➢ Add s + entry[0] = "ed". s = entry = "d".

* Step 8:

➢ s = "d" and k = 7.

➢ entry = "bd" (codeword 7 → "bd", created in Step 3).

➢ k is not in the base dictionary, it was added in a previous step.

➢ Output entry ("bd"). s is not empty.

➢ Add s + entry[0] = "db". s = entry = "bd".

* Step 9:

➢ s = "bd" and k = 3.

➢ entry = "c" (codeword 3 → "c").

➢ k is in the base dictionary.

➢ Output entry ("c"). s is not empty.

➢ Add s + entry[0] = "bdc". s = entry = "c".

* Step 10:

➢ s = "c" and k = 5.

➢ entry = "e" (codeword 5 → "e").

➢ k is in the base dictionary.

➢ Output entry ("e"). s is not empty.

➢ Add s + entry[0] = "ce". s = entry = "e".

* Step 11:

➢ s = "e" and k = 9.

➢ Since k = 9 has not been encountered before, this is a special case where entry is s + s[0] = "ee".

➢ Output entry ("ee"). s is not empty.

➢ Add s + entry[0] = "ee". s = entry = "ee".

* Step 12:

➢ s = "ee" and k = 2.

➢ entry = "b" (codeword 2 → "b").

➢ k is in the base dictionary.

➢ Output entry ("b"). s is not empty.

➢ Add s + entry[0] = "eeb". s = entry = "b".

* Step 13:

➢ s = "b" and k = 2.

➢ entry = "b" (codeword 2 → "b").

➢ k is in the base dictionary.

➢ Output entry ("b"). s is not empty.

➢ Add s + entry[0] = "bb". s = entry = "b".

* Final Step:

➢ s = "b" and k = EOF.

➢ Since k = EOF, the decoding process is complete.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **s** | **k** | **Item** | **Codeword** | **Output** |
| null | 1 | a | 1 | a |
| a | 2 | ab | 6 | b |
| b | 4 | bd | 7 | d |
| d | 3 | dc | 8 | c |
| c | 1 | ca | 9 | a |
| a | 5 | ae | 10 | e |
| e | 4 | ed | 11 | d |
| d | 7 | db | 12 | bd |
| bd | 3 | bdc | 13 | c |
| c | 5 | ce | 14 | e |
| e | 9 | ec | 15 | ca |
| ca | 2 | cab | 16 | b |
| b | 2 | bb | 17 | b |
| b | EOF | - | - | - |

Thus, decoder is “abdcaedbdcecabb”

2). Find encoder and decoder of LZW?

Given if we have: Input string: “cdaabbefacbdegfeeggf”

* Encoder: First, we find all different characters.
* There are “abcdefg”
* Build a base dictionary table:

|  |  |
| --- | --- |
| **Entry** | **Codeword** |
| a | 1 |
| b | 2 |
| c | 3 |
| d | 4 |
| e | 5 |
| f | 6 |
| g | 7 |

* s = 'c', c = 'd'
* s + c = 'cd' does not exist in the dictionary.
* Output codeword of s ('c' → 3).
* Add 'cd' to the dictionary. New codeword is 8.
* s = c = 'd'.
* s = 'd', c = 'a'
* s + c = 'da' does not exist in the dictionary.
* Output codeword of s ('d' → 4).
* Add 'da' to the dictionary. New codeword is 9.
* s = c = 'a'.
* s = 'a', c = 'a'
* s + c = 'aa' does not exist in the dictionary.
* Output codeword of s ('a' → 1).
* Add 'aa' to the dictionary. New codeword is 10.
* s = c = 'a'.
* s = 'a', c = 'b'
* s + c = 'ab' does not exist in the dictionary.
* Output codeword of s ('a' → 1).
* Add 'ab' to the dictionary. New codeword is 11.
* s = c = 'b'.
* s = 'b', c = 'b'
* s + c = 'bb' does not exist in the dictionary.
* Output codeword of s ('b' → 2).
* Add 'bb' to the dictionary. New codeword is 12.
* s = c = 'b'.
* s = 'b', c = 'e'
* s + c = 'be' does not exist in the dictionary.
* Output codeword of s ('b' → 2).
* Add 'be' to the dictionary. New codeword is 13.
* s = c = 'e'.
* s = 'e', c = 'f'
* s + c = 'ef' does not exist in the dictionary.
* Output codeword of s ('e' → 5).
* Add 'ef' to the dictionary. New codeword is 14.
* s = c = 'f'.
* s = 'f', c = 'a'
* s + c = 'fa' does not exist in the dictionary.
* Output codeword of s ('f' → 6).
* Add 'fa' to the dictionary. New codeword is 15.
* s = c = 'a'.
* s = 'a', c = 'c'
* s + c = 'ac' does not exist in the dictionary.
* Output codeword of s ('a' → 1).
* Add 'ac' to the dictionary. New codeword is 16.
* s = c = 'c'.
* s = 'c', c = 'b'
* s + c = 'cb' does not exist in the dictionary.
* Output codeword of s ('c' → 3).
* Add 'cb' to the dictionary. New codeword is 17.
* s = c = 'b'.
* s = 'b', c = 'd'
* s + c = 'bd' does not exist in the dictionary.
* Output codeword of s ('b' → 2).
* Add 'bd' to the dictionary. New codeword is 18.
* s = c = 'd'.
* s = 'd', c = 'e'
* s + c = 'de' does not exist in the dictionary.
* Output codeword of s ('d' → 4).
* Add 'de' to the dictionary. New codeword is 19.
* s = c = 'e'.
* s = 'e', c = 'g'
* s + c = 'eg' does not exist in the dictionary.
* Output codeword of s ('e' → 5).
* Add 'eg' to the dictionary. New codeword is 20.
* s = c = 'g'.
* s = 'g', c = 'f'
* s + c = 'gf' does not exist in the dictionary.
* Output codeword of s ('g' → 7).
* Add 'gf' to the dictionary. New codeword is 21.
* s = c = 'f'.
* s = 'f', c = 'e'
* s + c = 'fe' does not exist in the dictionary.
* Output codeword of s ('f' → 6).
* Add 'fe' to the dictionary. New codeword is 22.
* s = c = 'e'.
* s = 'e', c = 'e'
* s + c = 'ee' does not exist in the dictionary.
* Output codeword of s ('e' → 5).
* Add 'ee' to the dictionary. New codeword is 23.
* s = c = 'e'.
* s = 'eg', c = 'g'
* s + c = 'egg' does not exist in the dictionary.
* Output codeword of s ('eg' → 20).
* Add 'egg' to the dictionary. New codeword is 24.
* s = c = 'g'.
* s = 'gf', c = 'EOF'
* s + c = 'gf' does not exist in the dictionary (since it's EOF, we don't add to the dictionary).
* Output codeword of s ('gf' → 21).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **s** | **c** | **Item** | **Codeword** | **Output** |
| c | d | cd | 8 | 3 |
| d | a | da | 9 | 4 |
| a | a | aa | 10 | 1 |
| a | b | ab | 11 | 1 |
| b | b | bb | 12 | 2 |
| b | e | be | 13 | 2 |
| e | f | ef | 14 | 5 |
| f | a | fa | 15 | 6 |
| a | c | ac | 16 | 1 |
| c | b | cb | 17 | 3 |
| b | d | bd | 18 | 2 |
| d | e | de | 19 | 4 |
| e | g | eg | 20 | 5 |
| g | f | gf | 21 | 7 |
| f | e | fe | 22 | 6 |
| e | e | ee | 23 | 5 |
| eg | g | egg | 24 | 20 |
| gf | EOF | - | - | 21 |

Thus, encoder is : {3, 4, 1, 1, 2, 2, 5, 6, 1, 3, 2, 4, 5, 7, 6, 5, 20, 21}

➢ s='' and k=3.

➢ entry = 'c' (codeword 3→'c').

➢ k is in the base dictionary.

➢ Output entry ('c'). s is not empty.

➢ Add s+entry[0]='c' if s is not empty. s = entry = 'c'.

➢ s='c' and k=4.

➢ entry = 'd' (codeword 4→'d').

➢ k is in the base dictionary.

➢ Output entry ('d'). s is not empty.

➢ Add s+entry[0]='cd' if s is not empty. s = entry = 'd'.

➢ s='d' and k=1.

➢ entry = 'a' (codeword 1→'a').

➢ k is in the base dictionary.

➢ Output entry ('a'). s is not empty.

➢ Add s+entry[0]='da' if s is not empty. s = entry = 'a'.

➢ s='a' and k=1.

➢ entry = 'a' (codeword 1→'a').

➢ k is in the base dictionary.

➢ Output entry ('a'). s is not empty.

➢ Add s+entry[0]='aa' if s is not empty. s = entry = 'a'.

➢ s='a' and k=2.

➢ entry = 'b' (codeword 2→'b').

➢ k is in the base dictionary.

➢ Output entry ('b'). s is not empty.

➢ Add s+entry[0]='ab' if s is not empty. s = entry = 'b'.

➢ s='b' and k=2.

➢ entry = 'b' (codeword 2→'b').

➢ k is in the base dictionary.

➢ Output entry ('b'). s is not empty.

➢ Add s+entry[0]='bb' if s is not empty. s = entry = 'b'.

➢ s='b' and k=5.

➢ entry = 'e' (codeword 5→'e').

➢ k is in the base dictionary.

➢ Output entry ('e'). s is not empty.

➢ Add s+entry[0]='be' if s is not empty. s = entry = 'e'.

➢ s='e' and k=6.

➢ entry = 'f' (codeword 6→'f').

➢ k is in the base dictionary.

➢ Output entry ('f'). s is not empty.

➢ Add s+entry[0]='ef' if s is not empty. s = entry = 'f'.

➢ s='f' and k=1.

➢ entry = 'a' (codeword 1→'a').

➢ k is in the base dictionary.

➢ Output entry ('a'). s is not empty.

➢ Add s+entry[0]='fa' if s is not empty. s = entry = 'a'.

➢ s='a' and k=3.

➢ entry = 'c' (codeword 3→'c').

➢ k is in the base dictionary.

➢ Output entry ('c'). s is not empty.

➢ Add s+entry[0]='ac' if s is not empty. s = entry = 'c'.

➢ s='c' and k=2.

➢ entry = 'b' (codeword 2→'b').

➢ k is in the base dictionary.

➢ Output entry ('b'). s is not empty.

➢ Add s+entry[0]='cb' if s is not empty. s = entry = 'b'.

➢ s='b' and k=4.

➢ entry = 'd' (codeword 4→'d').

➢ k is in the base dictionary.

➢ Output entry ('d'). s is not empty.

➢ Add s+entry[0]='bd' if s is not empty. s = entry = 'd'.

➢ s='d' and k=5.

➢ entry = 'e' (codeword 5→'e').

➢ k is in the base dictionary.

➢ Output entry ('e'). s is not empty.

➢ Add s+entry[0]='de' if s is not empty. s = entry = 'e'.

➢ s='e' and k=7.

➢ entry = 'g' (codeword 7→'g').

➢ k is in the base dictionary.

➢ Output entry ('g'). s is not empty.

➢ Add s+entry[0]='eg' if s is not empty. s = entry = 'g'.

➢ s='g' and k=6.

➢ entry = 'f' (codeword 6→'f').

➢ k is in the base dictionary.

➢ Output entry ('f'). s is not empty.

➢ Add s+entry[0]='gf' if s is not empty. s = entry = 'f'.

➢ s='f' and k=5.

➢ entry = 'e' (codeword 5→'e').

➢ k is in the base dictionary.

➢ Output entry ('e'). s is not empty.

➢ Add s+entry[0]='fe' if s is not empty. s = entry = 'e'.

➢ s='e' and k=20.

➢ entry = 'eg' (codeword 20→'eg').

➢ k is in the base dictionary.

➢ Output entry ('eg'). s is not empty.

➢ Add s+entry[0]='ee' if s is not empty. s = entry = 'eg'.

➢ s='eg' and k=21.

➢ entry = 'gf' (codeword 21→'gf').

➢ k is in the base dictionary.

➢ Output entry ('gf'). s is not empty.

➢ Add s+entry[0]='egg' if s is not empty. s = entry = 'gf'.

* Decoder :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **s** | **k** | **Item** | **Codeword** | **Output** |
| null | 3 | c | - | c |
| c | 4 | cd | 8 | d |
| d | 1 | da | 9 | a |
| a | 1 | aa | 10 | a |
| a | 2 | ab | 11 | b |
| b | 2 | bb | 12 | b |
| b | 5 | be | 13 | e |
| e | 6 | ef | 14 | f |
| f | 1 | fa | 15 | a |
| a | 3 | ac | 16 | c |
| c | 2 | cb | 17 | b |
| b | 4 | bd | 18 | d |
| d | 5 | de | 19 | e |
| e | 7 | eg | 20 | g |
| g | 6 | gf | 21 | f |
| f | 5 | fe | 22 | e |
| e | 20 | ee | 23 | eg |
| eg | 21 | egg | 24 | gf |
| gf | EOF | - | - | - |

Thus, Decoder is “cdaabbefacbdegfeggf”