**Name: VEN THON**

**ID: e20191250**

**Group: I5-GIC-C**

Assignment Discussion07

1. Explain LZ77 compression algorithm?

The LZ77 compression algorithm is used to analyze input data and determine how to reduce the size of that input data by replacing redundant information with metadata.

* A match is encoded by a pair of numbers called a length-distance pair, which is equivalent to the statement.
* Each of the next length characters is equal to the characters exactly distance characters behind it in the uncompressed stream.
* The structure in which this data is held is called a sliding window.
* The encoder needs to keep this data to look for matches, and the decoder needs to keep this data to interpret the matches the encoder refers to.
* We can use LZ77 compression algorithm by:
* Set the coding position to the beginning of the input stream (first block and second block). Example: ababbbaaa (first block = 5; second block 2).
* Find the longest match in the window for the lookahead buffer. Example: ababbbaaa.
* If a match is found, output the pointer. Move the coding position (and the window) n bytes forward. Example: n = 2.
* If a match is not found, output a null pointer and the first byte in the lookahead buffer. Move the coding position (and the window) 1 byte forward. Example: n =1.
* If the lookahead buffer is not empty, return to step 2. Unless, finish searching.

1. By using LZ77 compression algorithm, find encoder and decoder from 2 different examples? You can choose your own string. Note: First block is more than second block!

Let **abbbccbbbcbaccdabbe** the string

We choose first 6 blocks: **abbbcc**

Next 4 blocks be the second: **bbbc**

Step 1:

* Compare 4 characters from first block with second block.

a b b b c c b b b c b a c c d a b

o “abbb” ≠ “bbbc” → move 1 character from first block

o “bbbc” = “bbbc” → match

a b b b c c b b b c b a c c d a b

6 5 4 3 2 1

We get: Codeword<5, 4, C(b)> (n=4)

Step 2: move n+4 (4+1=5) window at first block

* Keep taking 6 characters from first block and 4 characters from second block.

a b b b c c b b b c b a c c d a b

* Compare 4 characters from first block with second block.

c c b b b c b a c c d a b

o “ccbb” ≠ “bacc” → move 1 character from first block

o “cbbb” ≠ “bacc” → move 1 character from first block

o “bbbc” ≠ “bacc” → move 1 character from first block

o “So, remove 1 character at the end from second block.

o It rests 3 characters from second block: “dacc”.

* Compare 3 characters from first block with second block.

c c b b b c b a c c d a b

o “ccb” ≠ “bac” → move 1 character from first block

o “cbb” ≠“bac” → move 1 character from first block

o “bbb” ≠“bac” → move 1 character from first block

o “bbc” ≠“bac” → move 1 character from first block

o So, remove 1 character at the end from second block. o It rests 2 characters from second block: “ba”

* Compare 2 characters from first block with second block.

c c b b b c b a c c d a b

o “cc” ≠ “ba” → move 1 character from first block

o “cb” ≠ “ba” → move 1 character from first block

o “bb” = “ba” → move 1 character from first block