**Compress Algorithm**

Simply this algorithm is designed to compress the repeated sequence of data in the array. Since it’s a byte array (8 bit), but byte value is limited to 0 to 127 (7 bit), 8th bit can be used to denote the upcoming duplicate sequence.

This algorithm iterates over the complete array and set MSB bit as marking when it encounters a repeated sequence and followed by value containing the number of repetition follows.

**Example:**

Array before compression: {0x55, 0x55, 0x55, 0x55, 0x55, 0x55, 0x01, 0x0a, 0x0a, 0x0a, 0x07}

Array after compression: { 0xD5, 0x04, 0x01, 0x8a, 0x01, 0x07}

0XD5 -> MSB bit set in 0x55 to mark repeated sequence

0x04 -> Follow up bytes

0x8a -> MSB bit is set in 0x0a to mark repeated sequence

0x01 -> Follow up bytes

**Note:** If the repeated follow up byte count are more than 255, adjust byte is used

**Example:**

Array after compression: { 0xD5, 0xFF, 0xD5, 0x05, 0x01, 0x03}

Here 0x55 is repeated for 300 times, then

0xD5, 0xFF -> 0x55 repeated for 255 counts

0xD5, 0x05 -> 0x55 repeated for 5 counts

Totally 0x55 for 300 counts (Original data 300 bytes is reduced to 4 bytes efficiently)

**This algorithm is more efficient in reducing data size when there is more and more repeated sequence. Worst case possibility is when there are no repeated sequence data bytes remains same after conversion.**

**No addition temporary array is used while compression, hence reduced stack size. Compression is done by parsing the entire array only once and hence speed in execution.**

**Note : Through compress is the scope of this assignment, decompress is also done to unit test the compression source code**

|  |
| --- |
|  |
|  |  |