1 Decode

Algorithm 1 Decode algorithm

```
1:  procedure DECODE
2:      while input data exists do
3:          X ← get_bit(1)
4:          if X = 0 then
5:              Y ← get_bit(8)
6:              Output_token(Y)
7:          else
8:              offset ← Offset_get
9:              length ← Length_get
10:             Copy_ref(offset, length)
11:         end if
12:     end while
13: end procedure
```

2 Offset_get

Algorithm 2 Offset_get algorithm

```
1: procedure OFFSET_GET
2:     A ← Get_bit(1)
3:     if A = 0 then
4:         B ← get_bit(11)
5:         Use B to count offset value
6:     else
7:         C ← get_bit(7)
8:         Use C to count offset value
9:     end if
10:    Return offset
11: end procedure
```

3 Length_get

Algorithm 3 Length_get algorithm

```
1: procedure LENGTH_GET
2:     M ← get_bit(2)
3:     if M = 0b11 then
4:         N ← get_bit(2)
5:         if N = 0b11 then
6:             P ← get_bit(4)
7:             while P = 0b1111 do
8:                 num ← num + 15
9:                 P ← get_bit(4)
10:            end while
11:            length ← num + P + 8
12:        else
13:            Use N to count length value
14:        end if
15:    else
16:        Use M to count length value
17:    end if
18:    Return length
19: end procedure
```

4 Copy_ref

Algorithm 4 Copy_ref algorithm

```
1: procedure COPY_REF
2:     copy_num ← length
3:     Use offset to count read_address
4:     while copy_num ≠ 0 do
5:         Read history_data from history_ram base on the read_address
6:         Output_token(history_data)
7:         copy_num ← copy_num − 1
8:         read_address ← read_address + 1
9:     end while
10: end procedure
```

5 Output_token

Algorithm 5 Output_token algorithm

```
1: procedure OUTPUT_TOKEN(data)
2:     Put data to output buffer
3:     Put data to history_ram base on write_address
4: end procedure
```