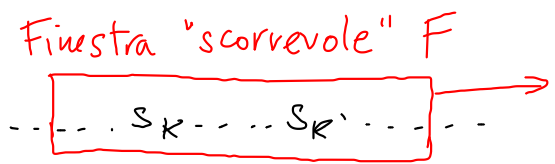
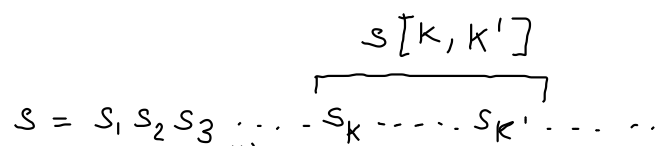
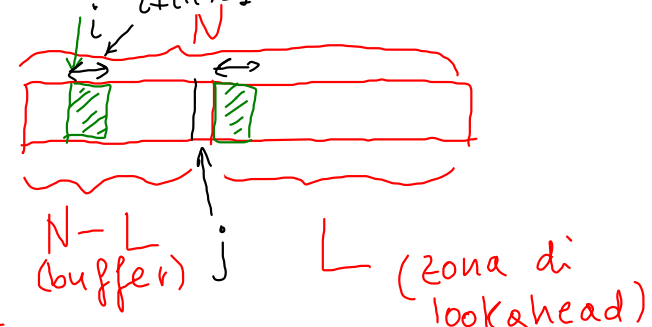


# CODICE DI ZIV-LEMPER (LZ77) → alla base dei formati .zip, .gzip

- Codice di tipo LV-B
- Stesso alfabeto per la sorgente e la codifica ( $A=B$ ;  $K=D$ )
- Codice universale (non utilizza la d.d.p. della sorgente)
- Asintoticamente ottimo
- Codifica di tipo strutturale (identifica ripetizione nella sequenza primaria)



Finestra F



Per ogni  $i=1, 2, \dots, N-L$  sia  $L(i)$  il più grande intero (\*) tale che

$$F[i, i + L(i) - 1] = F[j + 1, j + L(i)]$$

(\*) minore di  $L$

Parametri dell'algoritmo  
 $L$  (lunghezza max dei messaggi)  
 $N$  (lunghezza finestra)

$$j = N - L$$

Sia  $L(p) = \max_{1 \leq i \leq N-L} L(i)$

Chiamo :  $F[j+1, j+L(p)]$   
 $p$   
 $L(p)$

estensione riproducibile  
 puntatore della riproduzione ←  
 lunghezza della riproduzione ←

Esempio. Finestra di lunghezza  $N=8$  e lookahead  $L=5$  ( $\rightarrow j=N-L=3$ )

$F = 001|01011$

$i=1$	$\downarrow$ 0 0 1   0 1 0 1 1	$L(1) = 1$	} $L(p) = 4$ ( $p=2$ )
$i=2$	0 $\downarrow$ 0 1   $\downarrow$ 0 1 0 1 1	$L(2) = 4$	
$i=3$	0 0 $\downarrow$ 1   0 1 0 1 1 ↑	$L(3) = 0$	

La stringa  $s = s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_8 \dots$  sarà segmentata in messaggi  $m_1, m_2, m_3, \dots$

Ciascun messaggio  $m_i$  è codificato con una parola di codice di lunghezza costante ( $L_c$ ).

Parola di codice  $w_i$  associata a  $m_i$ :  $w_i = w_{i1} * w_{i2} * w_{i3}$

$w_{i1}$ : posizione del puntatore,  $p-1$  codificata in base  $K$  (lunghezza  $\lceil \log_K (N-L) \rceil$ )  
 $w_{i2}$ : lunghezza dell'estensione,  $L(p)$  codificata in base  $K$  (lunghezza  $\lceil \log_K L \rceil$ )  
 $w_{i3}$ : primo simbolo dopo la riproduzione (copiato) (lunghezza 1)

Algoritmo LZ77. Chiamiamo  $F_k$  la finestra al passo  $k$ -esimo dell'algoritmo ( $k$ -esimo messaggio)

1. Inizializza il contenuto di  $F_1$  con  $N-L$  simboli '0' e i primi  $L$  simboli di  $s$  ( $s[1, L]$ )

$$F_1 = 0^{N-L} * s[1, L]$$

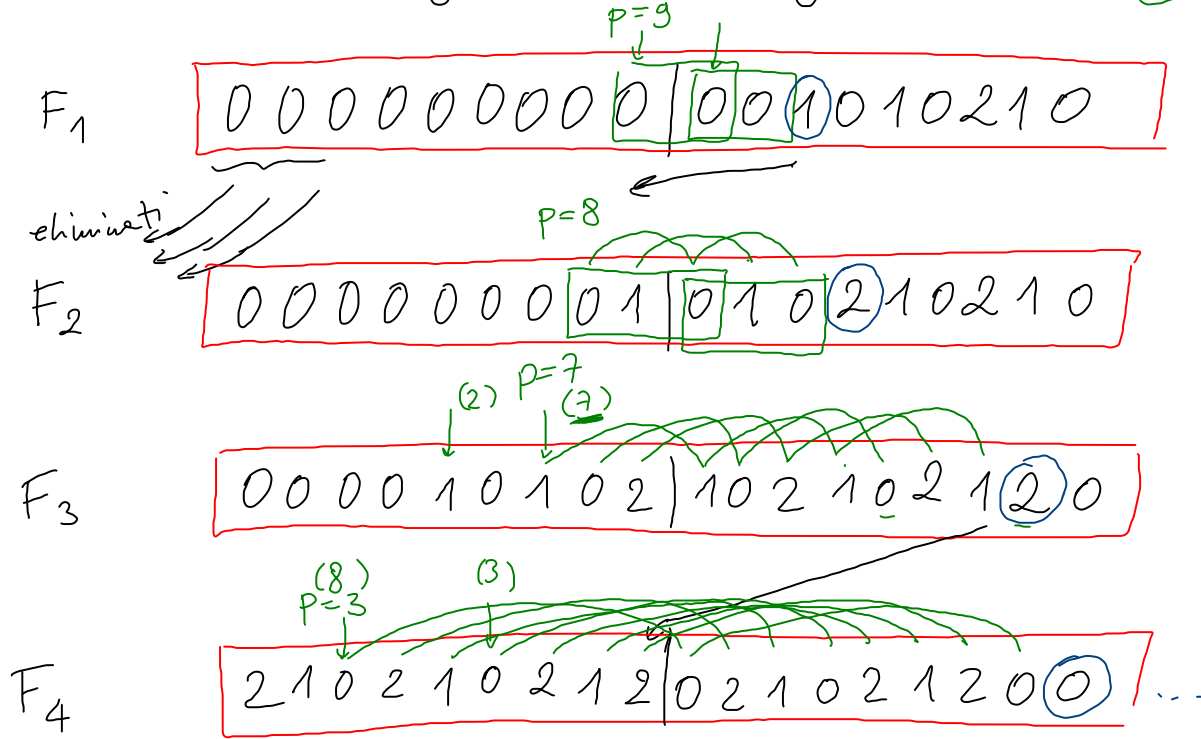
2. Noto  $F_k$  ( $k \geq 1$ ) sia  $m_k = F_k[\underbrace{\quad}_j, \underbrace{\quad}_j]$

3. Codifica  $m_k$  in una parola di codice  $w_k$  secondo lo schema già descritto

4. Aggiorna la finestra (trova  $F_{k+1}$ ) eliminando i primi  $L(p_k)+1$  simboli e facendone entrare altrettanti di nuovi.

Esempio.  $s = \overbrace{0010}^{m_1} \overbrace{1021}^{m_2} \overbrace{02102120}^{m_3} 021021200\$$  con parametri  $N=18, L=9; K=3$   
 $A=B=\{0,1,2\}$

$$L_c = \lceil \log_k(N-L) \rceil + \lceil \log_k L \rceil + 1 = 2 + 2 + 1 = 5$$



$p=9; p-1=8; L(p)=2$ ; muovo simbolo 1  
 $w_{11}=(22)_3 \quad w_{12}=(02)_3 \quad w_{13}=1 \rightarrow w_1=22021$

$p=8; p-1=7; L(p)=3$ ; muovo simbolo 2  
 $w_{21}=(21)_3 \quad w_{22}=(10)_3 \quad w_{23}=2 \rightarrow w_2=21102$

$p=7; p-1=6; L(p)=7$ ; muovo simbolo 2  
 $w_{31}=(20)_3 \quad w_{32}=(21)_3 \quad w_{33}=2 \rightarrow w_3=20212$

$p=3; p-1=2; L(p)=8$ ; muovo simbolo 0  
 $w_{41}=(02)_3 \quad w_{42}=(22)_3 \quad w_{43}=0$

$\rightarrow w_4 = 02220$

Codifica:  $w_1 * w_2 * w_3 * w_4$   
 $22021 | 21102 | 20212 | 02220$

Decodifica :  $w_1 = \underline{22} \underline{02} \textcircled{1}$

$$(22)_3 = 8 \rightarrow p = 8 + 1 = 9 \quad (02)_3 = 2 = L(p)$$

$F_1$ 

0000000000	001...
------------	--------

$F_2$ 

00000001	0102
----------	------

$F_3$ 

000010102	10210212
-----------	----------

$$w_2 = \underline{21} \underline{10} \textcircled{2}$$

$$(21)_3 = 7 \rightarrow p = 8 \quad (10)_3 = 3 = L(p)$$

$$w_3 = \underline{20} \underline{21} \textcircled{2}$$

$$(20)_3 = 6 \rightarrow p = 7 \quad (21)_3 = 7 = L(p)$$

$$s = \underbrace{001}_{m_1} \underbrace{0102}_{m_2} \underbrace{10210212}_{m_3}$$