

FILE COMPRESSI

.RAR .ZIP .7Z .ISO



PRINCIPALI ESTENSIONI

- 1) <https://winrar.it/>
- 2) <https://www.7-zip.org/download.html>
- 3) <https://www.winzip.com/it/>
- 4) <https://www.iso.org/home.html>



TIPOLOGIE DI ALGORITMI PER LA COMPRESSIONE

LOSSLESS	LOSSY
<p>ALGORITMI SENZA PERDITA O DEGRADO DELL'INFORMAZIONE ORIGINARIA (possibilità di decomprimere i dati compressi risalendo alla stessa identica sequenza di bit del file originale)</p>	<p>ALGORITMI CON PERDITA O DEGRADO DELL'INFORMAZIONE ORIGINARIA (impossibilità di decomprimere i dati compressi risalendo alla stessa identica sequenza di bit del file originale)</p>

The image features a light gray background with several thin, orange lines that intersect to form a large, irregular frame around the central text. The lines are positioned at various angles, creating a sense of dynamic movement and framing the content.

*CHE COSA SI INTENDE PER
ALGORITMO?!*

The background features several thin, light orange lines that intersect to form a series of irregular, abstract shapes, resembling a stylized frame or a network of connections.

ALGORITMI LOSSLESS

The background features several thin, light orange lines that intersect to form a series of irregular, abstract shapes, creating a modern, geometric frame around the central text.

RLE (*RUN LENGTH ENCODING*)

RLE (RUN LENGTH ENCODING)

BB NNN BBB NNN BBBBBB NN BBBBBB NNNNN BBBBB

RLE (RUN LENGTH ENCODING)

BB NNN BBB NNN BBBBNN NN BBBBNNNNNN BBBB -> 2B 3N 3B 3N 6B 2N 6B 5N 5B

RLE (RUN LENGTH ENCODING)

RLE (RUN LENGTH ENCODING)

fffff	fffff	000000	000000	000000	fffff	fffff
fffff	000000	000000	000000	fffff	fffff	fffff
fffff	fffff	fffff	000000	000000	fffff	fffff
fffff	fffff	fffff	fffff	000000	000000	000000
000000	000000	fffff	fffff	fffff	fffff	fffff

RLE (RUN LENGTH ENCODING)

fffff	fffff	000000	000000	000000	fffff	fffff
fffff	000000	000000	000000	fffff	fffff	fffff
fffff	fffff	fffff	000000	000000	fffff	fffff
fffff	fffff	fffff	fffff	000000	000000	000000
000000	000000	fffff	fffff	fffff	fffff	fffff

fffff fffff 000000 000000 000000 fffff fffff fffff 000000 000000 000000 fffff fffff fffff fffff fffff 000000 000000 fffff fffff
fffff fffff fffff fffff 000000 000000 000000 000000 000000 fffff fffff fffff fffff fffff

RLE (RUN LENGTH ENCODING)

fffff fffff 000000 000000 000000 fffff fffff fffff 000000 000000 000000 fffff fffff fffff fffff fffff fffff 000000 000000 fffff fffff
fffff fffff fffff fffff 000000 000000 000000 000000 000000 fffff fffff fffff fffff fffff

RLE (RUN LENGTH ENCODING)

```

fffff fffff 000000 000000 000000 fffff fffff fffff 000000 000000 000000 fffff fffff fffff fffff fffff fffff 000000 000000 fffff fffff
fffff fffff fffff fffff 000000 000000 000000 000000 000000 fffff fffff fffff fffff fffff

```

[illegible]

840 BIT = 105 BYTE

RLE (RUN LENGTH ENCODING)

fffff fffff 000000 000000 000000 fffff fffff fffff 000000 000000 000000 fffff fffff fffff fffff fffff fffff 000000 000000 fffff fffff
fffff fffff fffff fffff 000000 000000 000000 000000 000000 fffff fffff fffff fffff fffff

- Fissiamo un numero di bit per contare le ripetizioni -> esempio 3 bit -> al massimo consideriamo 8 ripetizioni adiacenti (2^3), se ne avessimo di più dovremmo spezzare le ripetizioni in due sequenze.

RLE (RUN LENGTH ENCODING)

fffff fffff 000000 000000 000000 fffff fffff fffff 000000 000000 000000 fffff fffff fffff fffff fffff fffff 000000 000000 fffff fffff
fffff fffff fffff fffff 000000 000000 000000 000000 000000 fffff fffff fffff fffff fffff

- Fissiamo un numero di bit per contare le ripetizioni -> esempio 3 bit -> al massimo consideriamo 8 ripetizioni adiacenti (2^3), se ne avessimo di più dovremmo spezzare le ripetizioni in due sequenze.

2 fffff 3 000000 3 fffff 3 000000 6 fffff 2 000000 6 fffff 5 000000 5 fffff

RLE (RUN LENGTH ENCODING)

fffff fffff 000000 000000 000000 fffff fffff fffff 000000 000000 000000 fffff fffff fffff fffff fffff fffff 000000 000000 fffff fffff
fffff fffff fffff fffff 000000 000000 000000 000000 000000 fffff fffff fffff fffff fffff

- Fissiamo un numero di bit per contare le ripetizioni -> esempio 3 bit -> al massimo consideriamo 8 ripetizioni adiacenti (2^3), se ne avessimo di più dovremmo spezzare le ripetizioni in due sequenze.

2 fffff 3 000000 3 fffff 3 000000 6 fffff 2 000000 6 fffff 5 000000 5 fffff

010 11111111111111111111111111111111 011 00000000000000000000000000000000 010 1111111111111111111111111111 110
00000000000000000000000000000000 110 1111111111111111111111111111 010 00000000000000000000000000000000 110 1111111111111111111111111111
101 00000000000000000000000000000000 101 1111111111111111111111111111

243 BIT = 30,375 BYTE

RLE (RUN LENGTH ENCODING)

```
010 11111111111111111111 011 00000000000000000000 010 11111111111111111111 110
00000000000000000000 110 11111111111111111111 010 00000000000000000000 110 11111111111111111111
101 00000000000000000000 101 11111111111111111111
```

↓ DECOMPRESSIONE ↓

[illegible]

RLE (RUN LENGTH ENCODING)

PRO	CONTRO
Estremamente semplice	Poco efficace su immagini con tante sfumature diverse
Efficace su immagini con pochi colori	Non efficace con sequenze poco ripetute
Efficace con sequenze molto ripetute	Non utilizzabile con file di testo
Buon rapporto di compressione	
Veloce nella decompressione	

The background features several thin, light orange lines that intersect to form a series of irregular, abstract shapes, resembling a stylized frame or a network of connections.

LZW (*LEMPEL, ZIV, WEICH*)

LZW (LEMPER, ZIV, WEICH)

ABABCBABAB

LZW (LEMPER, ZIV, WEICH)

ABABCBABAB

SIMBOLO	CODIFICA
A	
B	
C	

STABILISCO IL NUMERO DI BIT PER LA CODIFICA,
ESEMPIO: **3 BIT**

LZW (LEMPER, ZIV, WEICH)

ABABCBABAB

SIMBOLO	CODIFICA
A	000
B	001
C	010

LZW (LEMPER, ZIV, WEICH)

A BABCBABAB

SIMBOLO	CODIFICA
A	000
B	001
C	010

LZW (LEMPER, ZIV, WEICH)

ABABCBABAB

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011

LZW (LEMPER, ZIV, WEICH)

ABABCBABAB

000

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011

LZW (LEMPER, ZIV, WEICH)

ABABCBABAB

000

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011

LZW (LEMPER, ZIV, WEICH)

ABABCBABAB

000

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100

LZW (LEMPER, ZIV, WEICH)

ABABCBABAB

000 001

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100

LZW (LEMPER, ZIV, WEICH)

AB^ABCBABAB

000 001

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100

LZW (LEMPER, ZIV, WEICH)

AB^{AB}CBABAB

000 001

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100

LZW (LEMPER, ZIV, WEICH)

AB^{ABC}BABAB

000 001

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100
ABC	101

LZW (LEMPER, ZIV, WEICH)

AB^{AB}CBABAB

000 001 011

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100
ABC	101

LZW (LEMPER, ZIV, WEICH)

ABABCBABAB

000 001 011

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100
ABC	101

LZW (LEMPER, ZIV, WEICH)

ABABCBABAB

000 001 011

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100
ABC	101
CB	110

LZW (LEMPER, ZIV, WEICH)

ABABCBABAB

000 001 011 010

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100
ABC	101
CB	110

LZW (LEMPER, ZIV, WEICH)

ABABC**B**ABAB

000 001 011 010

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100
ABC	101
CB	110

LZW (LEMPER, ZIV, WEICH)

ABABCBA BAB

000 001 011 010

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100
ABC	101
CB	110

LZW (LEMPER, ZIV, WEICH)

ABABC**B**AB

000 001 011 010

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100
ABC	101
CB	110
BAB	111

LZW (LEMPER, ZIV, WEICH)

ABABCBA BAB

000 001 011 010 100

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100
ABC	101
CB	110
BAB	111

LZW (LEMPER, ZIV, WEICH)

ABABCBA**B**AB

000 001 011 010 100

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100
ABC	101
CB	110
BAB	111

LZW (LEMPER, ZIV, WEICH)

ABABCBA**B**AB

000 001 011 010 100

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100
ABC	101
CB	110
BAB	111

LZW (LEMPER, ZIV, WEICH)

ABABCBA**BAB**

000 001 011 010 100

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100
ABC	101
CB	110
BAB	111

LZW (LEMPER, ZIV, WEICH)

ABABCBABAB

000 001 011 010 100 111

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100
ABC	101
CB	110
BAB	111

LZW (LEMPER, ZIV, WEICH)

ABABCBABAB

000 001 011 010 100 111 = **18 BIT**

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100
ABC	101
CB	110
BAB	111

LZW (LEMPER, ZIV, WEICH)

ABABCBABAB

000 001 011 010 100 111 = 18 BIT

FILE NON COMPRESSO:

8 BIT x 10 CARATTERI = 80 BIT

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100
ABC	101
CB	110
BAB	111

LZW (LEMPER, ZIV, WEICH)

ABABCBABAB

000 001 011 010 100 111 = 18 BIT

FILE NON COMPRESSO:

8 BIT x 10 CARATTERI = 80 BIT

PRO VS CONTRO?

SIMBOLO	CODIFICA
A	000
B	001
C	010
AB	011
BA	100
ABC	101
CB	110
BAB	111

ALGORITMO DI HUFFMAN

