

Progetto compressore

Kuhn - Paoliello





LZ77 - Kuhn



Motivazioni

- “Semplicità”
- Combinabilità
- Ottimizzazioni

Funzionamento

... RTAAATARTTTATTARTARRRTARRR ...

... RTAAATARTTTATTARTARRRTARRR ...

... RTAAATARTTTATTARTARRRTARRR ...

... RTAAATARTTTATTARTARRRTARRR ...

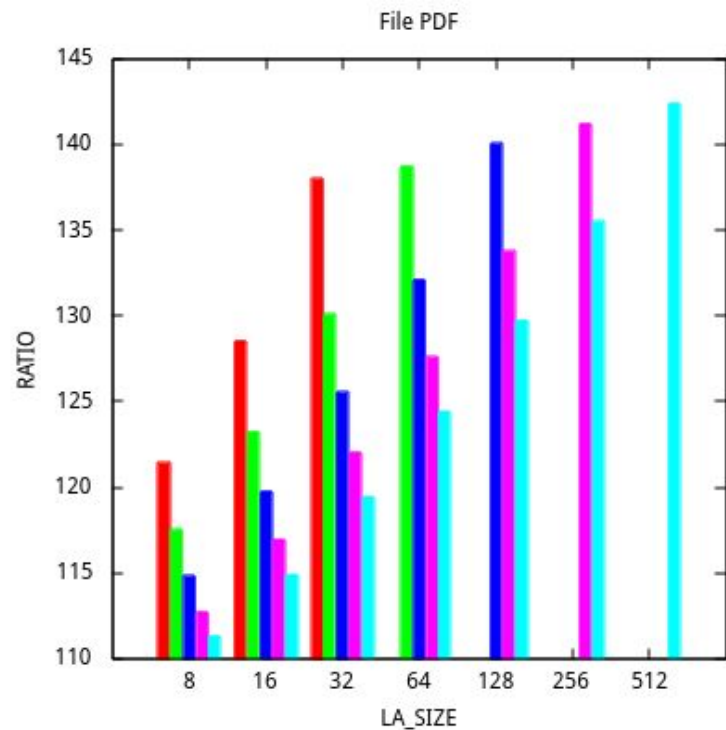
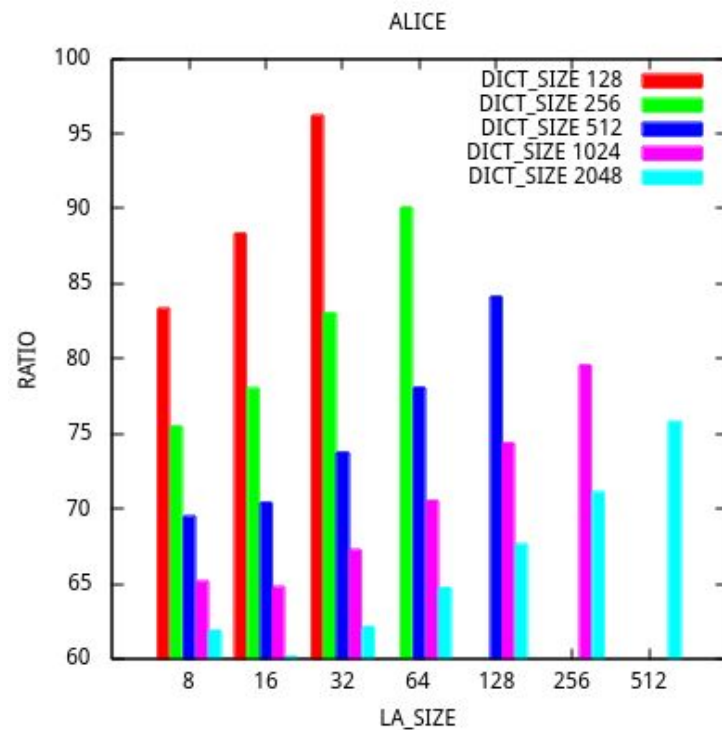
... RTAAATARTTTATTARTARRRTARRR ...

OFFSET = 4 LENGTH = 5 NEXT = R

Codificare l'informazione

- Dimensione dizionario: n_d
- Dimensione *look-ahead buffer*: n_{la}
- Bits per codificare offset: **$\text{lb}(n_d+1)$**
- Bits per codificare length: **$\text{lb}(n_{la})$**
- Buffer di scrittura

Risultati



Ottimizzazioni

- Sperimentate
 - Skip caratteri già letti
- Future
 - Maggiore compressione
 - Minore tempo di calcolo



Huffman - Paoliello



Motivazioni

- Funzionamento trasparente
- Modificabilità
- Combinabile con algoritmi con dizionario

Implementazione personale

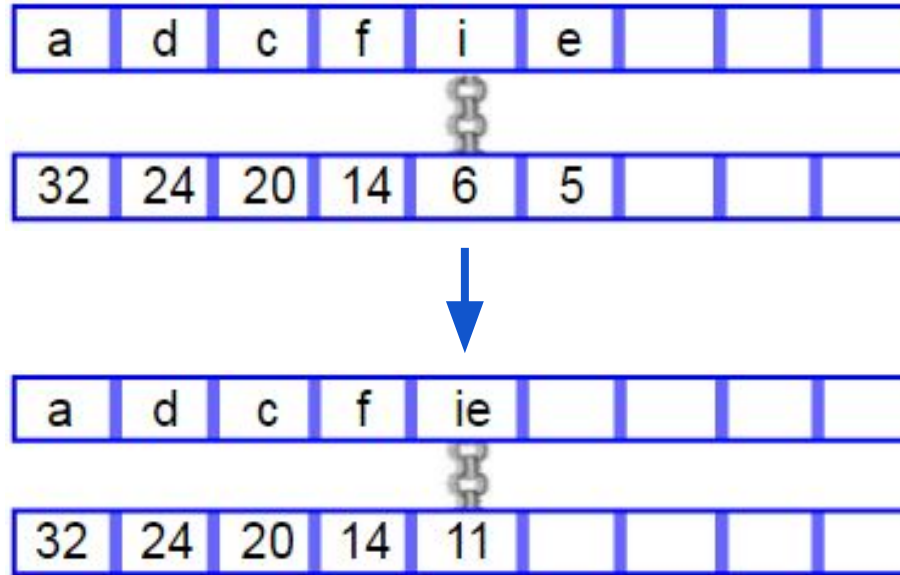
a	b	c	d	e	f	g	h	i
32	0	20	24	5	14	0	0	6



Raggruppa e ordina

a	d	c	f	i	e			
32	24	20	14	6	5			

Implementazione personale

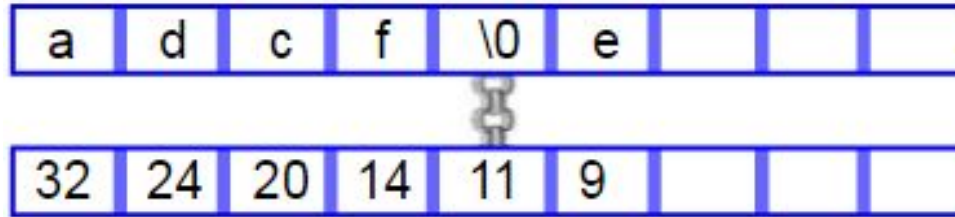


```
encodingStruct['i'].code+="0"  
encodingStruct['e'].code+="1"
```

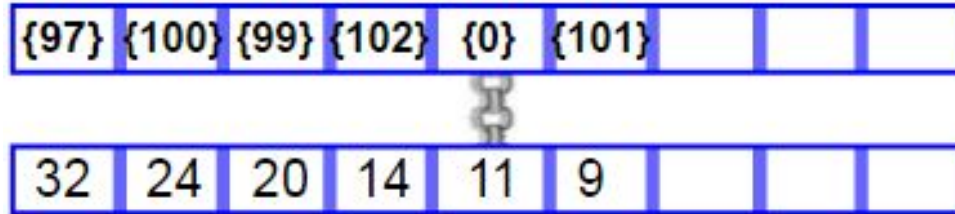
Reverse!

Implementazione personale

Il problema dello \0 (ASCII code: 0):



Soluzione:



Implementazione personale

Scrittura mappa canonizzata:

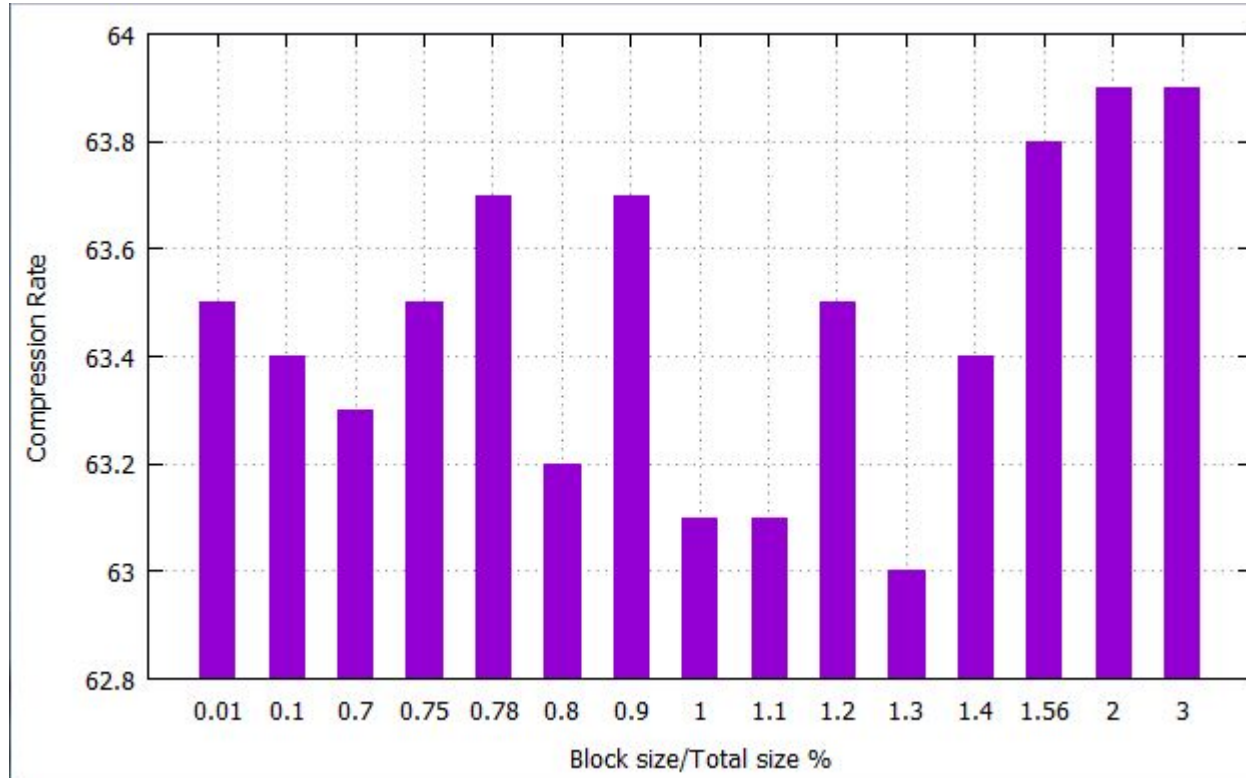
- Caratteri impliciti
- Minimizzare bit per descrivere la lunghezza delle codifiche, solitamente 5 bit



01110011010101010101111100001...

Mappa nel file compresso

Tentativo di ottimizzazione





Sviluppi futuri



Sviluppi futuri

- Huffman adattivo
- Combinazione “custom” Huffman-LZ77