Etudes

CS 1025 Computer Science Fundamentals I

Stephen M. Watt

University of Western Ontario

Arithmetic, Arrays, Input

- Write a program that takes lines of input consisting of numbers with optional decimal point
  and that computes and prints
  their average and standard deviation.
  The input terminates at the first line with no numbers.

average(x_1,...,x_n) = sum(x_i, i = 1..n)/ n

stdev(x_1,..., x_n) = sqrt(average((x – average(x_i))^ 2))

Arithmetic, Arrays, Input (breakdown)

- Write a program that takes one line of input containing numbers separated by spaces and computes their average.
- Modify your program so it reads until encountering a line with no numbers.
- Modify your program so it can handle numbers with decimal points.
- Modify your program so it saves the numbers in an array before computing the average.
- Modify your program so it also computes the standard deviation.

Data Compression

- Problem: To compress files of ASCII data including letters, digits, punctuation (!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~) and control characters 0 '\0' *NUL*, 07 '\a' *BEL*, 08 '\b' *BS*, 09 '\t' *HT*, 0A '\n' *LF*, 0B '\v' *VT*, 0C '\f' *FF*, 0D '\r' *CR*, 1B *ESC*, 20 *SPACE*, 7F *DEL*.

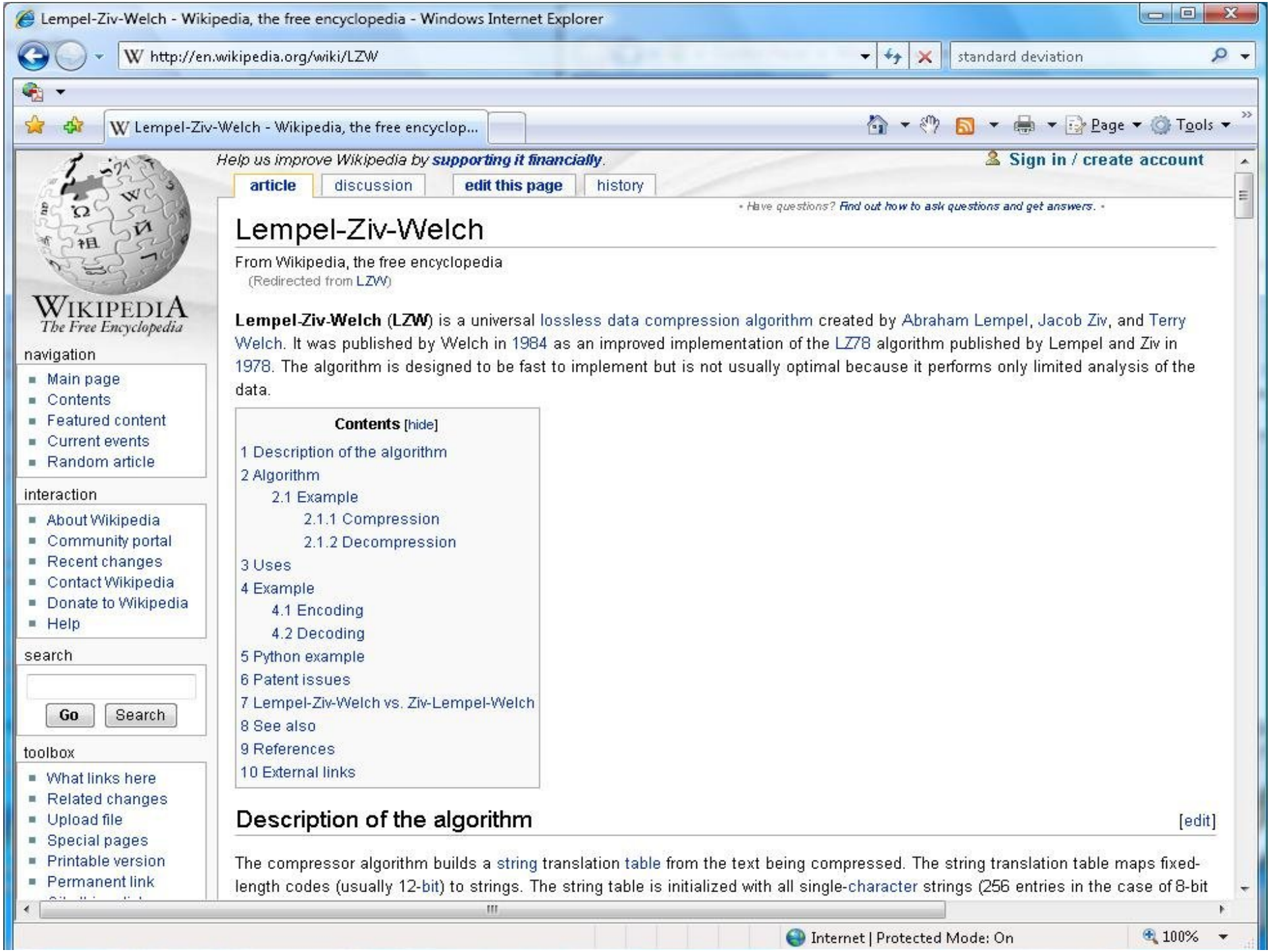
```
   2 3 4 5 6 7        30 40 50 60 70 80 90 100 110 120
  ------------       --------------------------------
0:   0 @ P ' p     0:    (  2  <  F  P  Z  d   n   x
1: ! 1 A Q a q     1:    )  3  =  G  Q  [  e   o   y
2: " 2 B R b r     2:    *  4  >  H  R  \  f   p   z
3: # 3 C S c s     3: !  +  5  ?  I  S  ]  g   q   {
4: $ 4 D T d t     4: "  ,  6  @  J  T  ^  h   r   |
5: % 5 E U e u     5: #  -  7  A  K  U  _  i   s   }
6: & 6 F V f v     6: $  .  8  B  L  V  '  j   t   ~
7: ' 7 G W g w     7: %  /  9  C  M  W  a  k   u  DEL
8: ( 8 H X h x     8: &  0  :  D  N  X  b  l   v
9: ) 9 I Y i y     9: '  1  ;  E  O  Y  c  m   w
A: * : J Z j z
B: + ; K [ k {
C: , < L \ l |
D: - = M ] m }
E: . > N ^ n ~
F: / ? O _ o DEL
```

Data Compression II

- There are many ways to compress data, including
  - Run-length encoding
  - Lempel-Ziv-Welch
  - Huffman encoding

Data Compression

- Start with the trivial compressor (no compression) to get the IO right.
- Then try the “negative compressor” that doubles each letter.
  GGeett tthhee ppooiinntt??
- Next do run-length encoding.
  Note that for codes < 0x20 we have used only 0, 0x7-0xD, 0x1B.
  We do not use the codes in the range 0x100-0x1FF at all.
- Determine a useful set of common interfaces and classes.
- Next do LZW compression.
- Revise common interfaces and classes as needed.


Lempel-Ziv-Welch - Wikipedia, the free encyclopedia - Windows Internet Explorer
http://en.wikipedia.org/wiki/LZW
standard deviation
Lempel-Ziv-Welch - Wikipedia, the free encyclop...
Page
Tools
Help us improve Wikipedia by supporting it financially.
Sign in / create account
article
discussion
edit this page
history
Have questions? Find out how to ask questions and get answers.
WIKIPEDIA
The Free Encyclopedia
navigation
Main page
Contents
Featured content
Current events
Random article
interaction
About Wikipedia
Community portal
Recent changes
Contact Wikipedia
Donate to Wikipedia
Help
search
Go
Search
toolbox
What links here
Related changes
Upload file
Special pages
Printable version
Permanent link
Lempel-Ziv-Welch
From Wikipedia, the free encyclopedia
(Redirected from LZW)
Lempel-Ziv-Welch (LZW) is a universal lossless data compression algorithm created by Abraham Lempel, Jacob Ziv, and Terry Welch. It was published by Welch in 1984 as an improved implementation of the LZ78 algorithm published by Lempel and Ziv in 1978. The algorithm is designed to be fast to implement but is not usually optimal because it performs only limited analysis of the data.
Contents [hide]
1 Description of the algorithm
2 Algorithm
2.1 Example
2.1.1 Compression
2.1.2 Decompression
3 Uses
4 Example
4.1 Encoding
4.2 Decoding
5 Python example
6 Patent issues
7 Lempel-Ziv-Welch vs. Ziv-Lempel-Welch
8 See also
9 References
10 External links
Description of the algorithm
[edit]
The compressor algorithm builds a string translation table from the text being compressed. The string translation table maps fixed-length codes (usually 12-bit) to strings. The string table is initialized with all single-character strings (256 entries in the case of 8-bit
Internet | Protected Mode: On
100%

Ceaser's Cryptography

- Substitution ciphers. A -> N, B -> R, ...
- Broken using combination of
  - frequency information,
  - dictionaries
  - exhaustive search

- In English: E T A O I N S H R D L U... (ex space, punctuation)
- Bigrams, Trigrams: TH THE

- Problems:
  - Construct a frequency table for n-grams (n=1,2) from a corpus (e.g. some Gutenberg texts)
  - Given a cipher-text propose inverse substitutions.
  - Language guessing.