

O O bet365

LZW (Lempel-Ziv-Welch) é um algoritmo de compressão de dados sem perdas, desenvolvido por Abraham Lempel e Jacob Ziv em 1978. A sigla "LZW" significa "Lempel-Ziv-Welch", homenagem a seu criador e o cientista de computação Terry Welch, que desenvolveu uma implementação eficiente do algoritmo.

O algoritmo funciona construindo uma tabela de cadeias de caracteres medida que lê a entrada. Inicialmente, a tabela contém apenas as cadeias de caracteres vazias e os caracteres individuais. Para cada caractere lido, o algoritmo procura a cadeia de caracteres mais longa na tabela que é um prefixo da cadeia de entrada atual e emite a próxima entrada como um par (comprimento da cadeia prefixo, novo caractere). Em seguida, a tabela é atualizada adicionando a nova cadeia de caracteres formada pelo prefixo e o novo caractere.

O processo continua até que a entrada seja esgotada, momento em que o algoritmo emite o último par e termina. O resultado é uma sequência de pares (comprimento, caractere) que representam a entrada original comprimida.

A descompressão funciona basicamente da mesma forma, construindo a tabela medida que lê a entrada. Inicialmente, a tabela contém apenas as cadeias de caracteres vazias e os caracteres individuais. Para cada par (comprimento, caractere) lido, o algoritmo constrói a cadeia de caracteres prefixada pelo comprimento lido e adiciona o caractere ao final da cadeia. Em seguida, a tabela é atualizada adicionando a nova cadeia de caracteres formada.

Um exemplo prático: imagine a entrada "abcabcabc". O algoritmo emitirá os pares (1, 'a'), (1, 'b'), (1, 'c'), (2, 'ab'), (2, 'bc'), (3, 'abc').

Um exemplo de aplicação: o algoritmo LZW é usado em formatos de arquivo como GIF e TIFF. Também é usado em protocolos de rede como o protocolo de compressão de dados do protocolo de transferência de hipertexto (HTTP).

Um exemplo de implementação em Python:

```
def compress(LZW, s):
    table = {}
    for c in s:
        table[c] = c
    i = 0
    while i < len(s):
        j = i + 1
        while j < len(s) and s[i:j] in table:
            j += 1
        table[s[i:j]] = s[i:j]
        yield (j - i, s[i])
        i = j
```

Para mais informações, consulte o artigo "LZW" no livro "Algoritmos de Compressão de Dados" de Mark Nelson. Também há uma página sobre o algoritmo LZW no site do Python.org.