

Huffman Kodlama

Dosya Sıkıştırma: Dosya sıkıştırmaya minimum alana max veri sığdırmak için kullanılır. Daha yoğun bir gösterimle temsil edilen bitlerin sayısı azaltılmaktadır.

Dezavantajları:

- Kodlama için fazladan maliyet ister
- Kodun çözülmesi gerekir, bu da karmaşıklığı artırır.

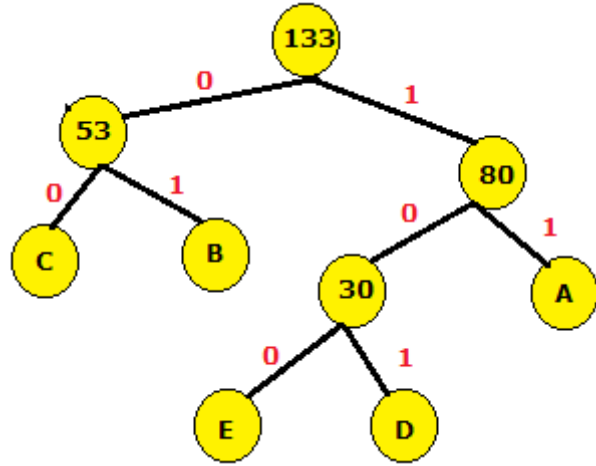
Huffman kodlaması kayıpsız sıkıştırma algoritmasıdır. Değişken uzunlukta kodlamaya sahiptir. Bu kodlamanın uzunluğu veri setindeki karakterlerin tekrar sayısına bağlıdır.

Huffman algoritmasında bir veri setinde daha çok rastlanan bir karaktere daha az uzunluktaki kod, daha az rastlanan karaktere daha büyük uzunluktaki kodla temsil edilmesine dayanır.

Her karakterin ne kadar sıklıkla tekrar ettiğini gösteren tabloya frekans tablosu denir. Karakterler her zaman ağacın yapraklarında tutulur.

Örnek:

Karakter	Frekans
A	50
B	28
C	25
D	20
E	10
Gereken Bit Sayısı (1 char=8 Bit)	1064 Bit



Karakter	Frekans	İkili Kodu
A	50	11
B	28	01
C	25	00
D	20	101
E	10	100
Gereken Bit Sayısı		296 Bit

LempelZiv Kodlama

Bilgisayar bilimlerinde kullanılan sıkıştırma oranı en yüksek algoritmalarından biridir. Algoritma bir mesajda, harf harf ilerleyerek mümkün olan en fazla harfi içeren kelimeyi sözlüğe eklemeyi amaçlamaktadır.

Unix ortamındaki zip-unzip komutları bu kodlamayı kullanmaktadır.

Örnek:

AAABBCBCDDDEAB karakter katarını Lempel Ziv algoritmasına göre kodlayınız.

Karakter setini daha önce hiç görülmemiş en küçük parçalar ayır

A AA B BC BCD D DE AB

İndeks değerleri verelim

0=NULL string

1 2 3 4 5 6 7 8
A AA B BC BCD D DE AB

Sözlük:

Çıkış	İndeks	Katar
(0,A)	1	A
(1,A)	2	AA
(0,B)	3	B
(3,C)	4	BC
(4,D)	5	BCD
(0,D)	6	D
(6,E)	7	DE
(1,B)	8	AB

Aşağıdaki sıkıştırılmış mesajda her kod sözcüğü bir karakter ve bir integer içermektedir. Karakter kısmı bellekte 1 byte (8 bit) yer işgal eder. Integer kısmı için ise örneğin 0 ve 1 için 1 bit, 2 ve 3 için (10, 11) 2 bit, 4 ve 6 (100, 110) için 3 bit yer işgal edecektir.

Sıkıştırılmış Mesaj: (0,A) (1,A) (0,B) (3,C) (4,D) (0,D) (6,E) (1,B)

Uzunluk(Bit): (1+8) (1+8) (1+8) (2+8) (3+8) (1+8) (3+8) (1+8)

Toplam: 9+9+9+10+11+9+11+9 = 77 bit

Hazırlayan
Arş. Gör. M. Fatih ADAK