

Huffman Kodu

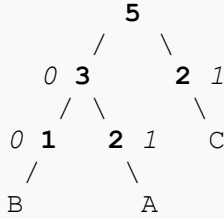
Huffman Kodu, [Bilgisayar biliminde](#), [veri sıkıştırması](#) için kullanılan, bir [entropi kodlama algoritmasıdır](#). [David A. Huffman](#) tarafından 1952 yılında geliştirilmiştir.

Huffman'ın algoritması, her [sembol](#) (veya [karakter](#)) için özel bir [kod](#) üretir. Bu kodlar ([ikilik sistemdeki](#) 1 ve 0'lardan oluşan) [bit](#) haritası şeklindedir. [Veri](#) içerisinde en az kullanılan karakter için en uzun, en çok kullanılan karakter için ise en kısa kodu üretir.

Huffman tekniği günümüzde tek başına kullanılmaz. [LZW](#), [RLE](#) gibi yöntemlerle birlikte kullanılır.

Teknik

Huffman'ın algoritması, veri içerisindeki karakterlerin kullanım sıklığına ([frekans](#)) göre bir ağaç oluşturur. Ağacın en tepesinden aşağıya doğru ilerlerken sola ayrılan dal için 0, sağa ayrılan dal için 1 kodu verilir.



Yukarıda koyu rakamlar karakter sayısını (kullanım sıklığı-frekans) gösterir, eğik rakamlar ise bit kodlarını gösterir. Bu ağaç "ABC" karakterlerinden oluşan bir veri kümesi için üretilmiştir.

Ağaca göre karakterler için bit haritaları şu şekildedir:

B: 00
A: 01
C: 1

Oluşturulan bit haritaları karakterlerin veri içerisindeki konumlarına göre yerleştirilir. Ortaya çıkan bit haritası sıkıştırılmış veridir.

Örneğin; "BAACC" verisi elde edilen bit haritalarına göre yeniden düzenlenirse:

00 01 01 1 1 = 00010111 = 17h

Yani %80 oranında bir sıkıştırma elde edilmiş olur.

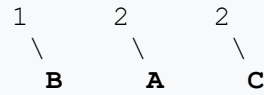
Ağacın oluşturulması

İlk önce karakterlerin frekansları (kullanım sıklıkları) hesaplanmalıdır.

Örneğin, elimizdeki veri "**BAACC**" olsun,

B: 1
A: 2
C: 2

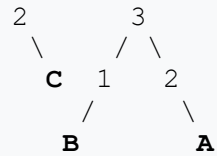
B:1 **A**:2 **C**:2



En küçük iki frekans toplanır ve frekans tablosu yeniden düzenlenir,

B:1 + **A**:2 **C**:2

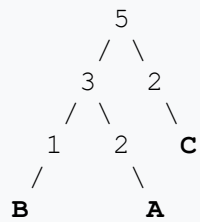
C:2 **BA**:3



Tek bir ağaç oluşturulana kadar sürekli en küçük frekanslar toplanır,

C:2 + **BA**:3

CBA:5



Dezavantajları

Huffman algoritması az sayıda karakter çeşidine sahip ve büyük boyutlardaki verilerde çok kullanışlı olabilir. Fakat oluşturulan ağacın sıkıştırılmış veriye eklenmesi zorunludur. Bu da sıkıştırma verimini düşürür. **Adaptive Huffman** gibi teknikler bu sorunu halletmek için geliştirilmişlerdir.