**Heap Sort**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

* İki tür heap ağacı vardır: max heap(büyükten küçüğe) ve min heap(küçükten büyüğe)
* Min heapte ağaç: ebeveyn her zaman çocuklarından küçük olmalıdır. Kökte her zaman ağacın en küçük değeri vardır.
* Max heapte de tam tersi

Min heap:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

* Mutlaka dizi kullanılmalı. Bağıl olarak yaparsan yavaşlarsın çünkü her zaman en sağ aşağı düğüm siliniyor. Diz,de de anlık erişim var

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Ebevyn çocuklarından büyük olursa büyükten küçüğe, küçük olursa küçükten büyüğe sıralanır

**Veri Sıkıştırma**

* Kayıplı sıkıştırma da orijinal haline ulaşamazsın.
* Kayıpsız sıkıştırmanın en başarılı olduğu algoritma metin sıkıştırmadır
* Metin dosyalarının kaybolmaması gerekir
* Metin sıkıştırmada önemli olan sıkıştırma düzeyi/oranı

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, alkol içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Lempelziz Algoritması**

* Önce metni okumaya başlar

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Ne kadar yer kaplar? Her karakter 1 byte yer kaplar.

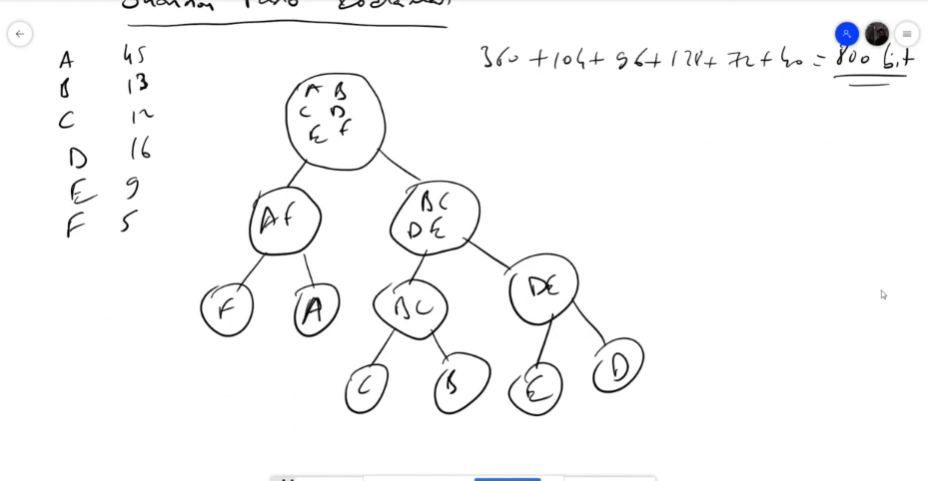
metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

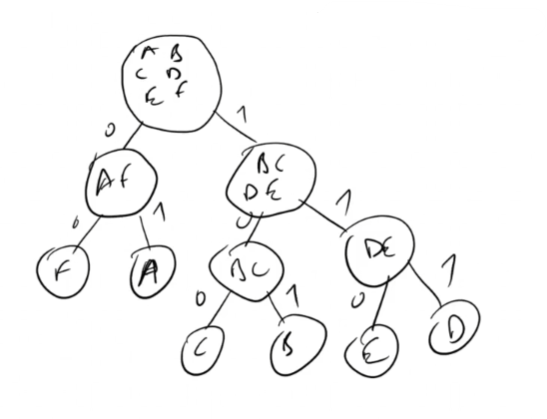
Karşı taraf sözlüğü bilmediği için sözlüğü de göndermen lazım. O da yer kaplar. Şimdilik sözlüğün kapladığı yeri hesaba katmıyoruz. Bu metin 80 bit yer kaplar. Başlangıçta ise her harf bir byte dı 1 byte = 8bit 20 karakter vardı 20x8 den 160 bit yer kaplıyordu

**Shannon Fano Algoritması**

* Bütünden parçalara doğru gider
* Önce dosyayı tarar hangi karakterden kaç adet var bulur
* NOT: sıkıştırmak için harcayacağımız sürenin bir önemi yok.
* Her zaman küçüğü sola yerleştir



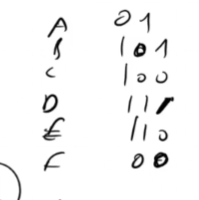
Bilgi yaprklardadır. Bc diye bir kodlama olmaz tek harf kalana kadar bölünür. Sola gittikçe 0 sağa gittikçe 1 yerleştir



Ağaç kullnılmasının sebebi benzersiz başlama bulmaktır

Çok sık kullanılanlara düşük bit, az kullanılanlara yüksek bit verilir. Mantık budur. Küçükleri olabildiğince yaprağa yaklaştırmalısın. Küçükler daha aşağıda olmalı

Kodlama kökten başlanarak yapılır



Yukarıdaki biraz olmadı gibi düzeltelim:

