**amblem, logo, simge, sembol, ticari marka içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**T.C**

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI**

**ÖDEV KONUSU**

**PROGRAMLAMA LAB. TELAFİ ÖDEVİ**

**Hazırlayan**

**UTKU EMRE ERMİŞ**

**220501034**

**DERS SORUMLUSU**

**ARŞ. GÖR. ŞEVVAL ŞOLPAN ARŞ. GÖR. ERAY DURSUN**

1. **ÖDEVİN KONUSU**

Txt dosyasındaki kelime sayısını “heap” yapısı ile sıralayan program

1. **HEAP NEDİR?**

Yığın ağacı(heap) bilgisayar bilimlerinde özellikle sıralama amacıyla çokca kullanılan bir veri yapısıdır. Bu veri yapısı üst düğümün (atasının) alt düğümlerden (çocuklarından) her zaman büyük olduğu bir ikili ağaç (binary tree) şeklinde düşünülebilir.

**ÖR**: daire içeren bir resim

Açıklama otomatik olarak oluşturuldu

Yukarıdaki ikili ağaçta dikkat edilirse ağaç dengeli bir şekilde sırayla doldurulmuştur. Buna göre yeni bir eleman daha eklenmesi durumunda ağacın eleman sayısı 7’ye yükselecek ve sayıların yeri değişmekle birlikte yeni eleman şu andaki 18 sayısını içeren düğümün sağına gelecektir.

Yığın ağaçları her zaman sırayla dolar ve sırayla boşalır.

1. **ÖDEVİN YAPILIŞI**
2. **“WordHeapNode” sınıfı bir kelimenin ve onun sayısının saklandığı bir düğümü temsil eder. Bu sınıfın iki özelliği vardır: “word ve count”.**

**“word”: Bu özellik, düğümün temsil ettiği kelimeyi saklar.**

**“count”: Bu özellik, kelimenin metinde kaç kez geçtiğini saklar.**

**Bu sınıf ayrıca “\_\_lt\_\_” adında bir metoda sahiptir. Bu metod, iki “WordHeapNode” örneğinin karşılaştırılmasını sağlar. Eğer iki düğümün sayısı aynıysa, kelimeler alfabetik sıraya göre karşılaştırılır. Aksi takdirde, sayılarına göre karşılaştırılır. Bu metod, yığın veri yapısının düğümleri doğru bir şekilde sıralamasını sağlar. Bu sayede, en sık veya en az kullanılan kelimeleri belirlemek mümkün olur.**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

1. **“count\_words” fonksiyonu bir metindeki kelimelerin frekansını hesaplar.**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Fonksiyon, bir dosya yolunu parametre olarak alır.**

**Dosya, okuma modunda ('r') ve “UTF-8” kodlamasıyla açılır.**

**Dosyanın tüm içeriği okunur ve küçük harfe çevrilir.**

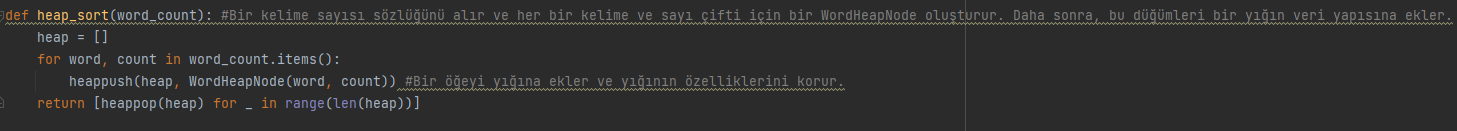
**“re.findall” fonksiyonu, metindeki tüm kelimeleri bulmak için kullanılır. “\b\w+\b” ifadesi, bir kelimenin başlangıcını ve sonunu belirler.**

**“defaultdict(int)” kullanılarak bir sözlük oluşturulur. Bu sözlük, her kelimenin kaç kez geçtiğini saklar.**

**Metindeki her kelime için, kelimenin sayısı bir artırılır.**

**Son olarak, kelime sayısı sözlüğü döndürülür.**

1. **“heap\_sort” fonksiyonu bir kelime sayısı sözlüğünü alır ve her bir kelime ve sayı çifti için bir “WordHeapNode” oluşturur. Daha sonra, bu düğümleri bir yığın veri yapısına ekler.**

****

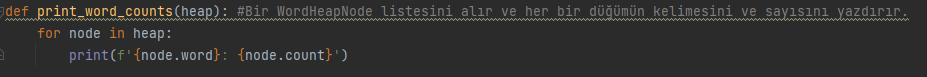
**Fonksiyon, bir kelime sayısı sözlüğünü parametre olarak alır.**

**Boş bir yığın oluşturulur.**

**Sözlükteki her kelime ve sayı çifti için, bir “WordHeapNode” oluşturulur ve bu düğüm yığına eklenir. “heappush” fonksiyonu, bir öğeyi yığına ekler ve yığının özelliklerini korur.**

**Son olarak, yığındaki tüm düğümler çıkarılır ve bir liste olarak döndürülür. “heappop” fonksiyonu, yığından en küçük öğeyi çıkarır ve yığının özelliklerini korur.**

1. **“print\_word\_counts” fonksiyonu bir “WordHeapNode” listesini alır ve her bir düğümün kelimesini ve sayısını yazdırır.**

****

**Fonksiyon, bir “WordHeapNode” listesi alır.**

**Liste üzerinde bir döngü başlatılır ve her bir düğüm için, düğümün kelimesi ve sayısı yazdırılır.” f'{node.word}: {node.count}'” ifadesi, bir “f-string” ifadesidir ve düğümün kelimesini ve sayısını bir “string” içinde biçimlendirir.**

1. **“main” fonksiyonu programın ana giriş noktasıdır ve diğer tüm fonksiyonları çağırır.**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Fonksiyon, kullanıcıdan bir txt dosyasının yolunu alır.**

**“count\_words” fonksiyonunu çağırarak bu dosyadaki kelimelerin sayısını hesaplar.**

**“heap\_sort” fonksiyonunu çağırarak bu sayıları sıralar.**

**“print\_word\_counts” fonksiyonunu çağırarak sonuçları yazdırır.**

1. **KAYNAKÇA**

[**https://bilgisayarkavramlari.com/2008/08/09/yigin-agaci-heap/**](https://bilgisayarkavramlari.com/2008/08/09/yigin-agaci-heap/)

1. **GİTHUB HESABI**

**https://github.com/UtkuEmreErmis**