

**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BLM0111**

**ALGORİTMA VE PROGRAMLAMA DERSİ PROJE RAPORU**

**Adı-Soyadı:** Erdal NAYIR **Adı-Soyadı:** Yakup ABACI

**Numarası:** 19360859049 **Numarası:** 19360859045

Github repository linki : https://github.com/ErdalNayir/Algoritma\_ve\_programlama\_proje/

Proje ödevine başlamadan önce beraber bize verilen “algo\_prog\_Proje” adlı PDF dosyasını inceleyip konuyu tam bir şekilde anlamaya çalıştık. PDF içinde bulunan “bigram”, “trigram” ve frekans hesaplama terimlerini bilmediğimiz için internet üzerinden çeşitli kaynakları taradık. İnternet dışında öğretim görevlisi hocalarımıza Telegram üzerinden bu terimlerin ne olduğunu ve frekans değerlerinin hesaplanmasını nasıl yapacağımızı sorup elde ettiğimiz bilgiler ile ödevimize yapmaya başladık.

İlk olarak bizden istenen ilk görevi (metnin harf dışındaki karakterlerden temizlenmesini) ikimiz ayrı olarak yazıp ve ikimizin de başarılı olduğu yerleri birleştirip tek bir kod haline dönüştürmeyi planladık. Bu fikir üzerine devam ederek ikimizde 2 farklı kod yazdık. Sonrasında bunları birleştirmeye çalıştığımızda zorlandığımızı ve kodlarımızın uzadığını fark ettik. Kodumuzun verim açısından sıkıntılı olacağını düşündük ve kodlarımızı beraber yazıp hatalarımızı daha kolay fark edeceğimizi ve daha performanslı çalışacağını düşünüp ayrı ayrı yapmak yerine her aşamasında beraber yaparak ödevi tamamlama kararı aldık.

İlk karşılaştığımız zorluk karakter temizlemede nokta(.) virgül (,) vb. karakterleri ve sayıları boşluk karakterine dönüştürmekti. Bunun için de iç-içe koşul ifadeleri kullandık. Yazdığımız kodlarda bu karakterleri silebiliyorduk fakat yazılan büyük harfleri de boşluk karakteri olarak yazdırıyordu. Sıkıntımızın ise koşulumuzda else karakterini yanlış olarak kullanmamızdı. Bu problemi fark ettikten sonra kodumuzu çalıştırdığımızda sıkıntısız çalıştığını gördük. Ve bir sonraki bizden istenen aşamaya geçtik.

İkinci olarak istenen görevimiz ise taslak kodda verilen tüm bigramların, kullanıcının girdiği metinde frekanslarının bulunmasıydı. Bu aşamada kafamızı karıştıran olay verilen bigramların karakter dizisi şeklinde değil de matris şeklinde verilmesiydi. Bu sıkıntımızı gidermek için internette aramalar yaptık.[2] Bir sitede aradığımızı açıklayan bir tablo gördük. Sonrasında bizden istenen görevi yapmak için ilerlememiz gereken yolu düşündük. İlk başta verilen her bigramın metinde kaç kez bulunduğunu bulan ve bunu her bulduğunda bir değişkeni arttıran bir iç-içe döngü yazdık. Bu kısımda karşılaştığımız bir diğer sıkıntı ise değişkenin üzerine önceki bulduğumuz bigram sayısının da eklenmesiydi. Burada değişkenimizi her bir sonraki bigrama geçtiğinde sıfırlanmayıp bize toplam bigram sayısını vermesiydi. Bunun çözümü için her döngünün sonunda sıfırlanan yeni bir değişken tanımladık. Bulduğumuz her bigramları ekrana yazdıran kodumuzu tamamladıktan sonra frekansını bulma kısmına geçtik. Bu konuda “Arş. Gör. Muhammed Ömer Faruk SELVİ” hocamıza sorup hesaplanmasının nasıl yapıldığını “Telegram” üzerinden öğrendik. Daha sonrasında her bir bulduğumuz bigramın toplam bigrama oranını hesaplayan kodumuzu yazdıktan sonra test ettik. Bulduğumuz bu frekans değerlerini ileride kullanacağımız bir dizi içine kaydettik bu aşamayı da tamamladıktan sonra bizden istenen diğer işleme geçtik.

Üçüncü olarak istenen ise trigramların hesaplanmasıydı. Bu kısım için yukarıda bigramları hesaplarken farklı olarak koşullar içine yeni bir koşul daha ve iç-içe döngümüzün içine de yeni bir döngü daha eklemek oldu ve yine bulduğumuz bu frekans değerlerini ileride kullanabilmek için kendi oluşturduğumuz diziye kaydettik.

Bu sırada aklımıza takılan bir diğer soru ise kullanıcının girdiği metinde büyük harf şeklinde bigramın ya da trigramın hesaplanacakmıydı. Biz buraya kadar olan işlemlerimizde sadece küçük harfli olan bigramlar ve trigramlar da hesaplamak üzerine kodlarımızı yazmıştık. Bu sorumuzu **“**Arş. Gör. Esma İBİŞ” hocamıza Telegram üzerinden danıştık. Hocamız bunu sorunumuz için bize verdiği tavsiye sonrasında büyük harfleri küçük harflere dönüştüren, bigram ve trigramların sayısını bulan kodumuzun öncesine bir döngü daha ekledik. Bu sayede artık büyük harfle yazılan bigram veya trigramların sayısını ve frekansını doğru bir şekilde hesaplamayı tamamladık. Ve bir sonraki yapmamız gereken işleme geçtik.

Bizden istenen dördüncü işlem ise Öklid uzaklığını hesaplamaktı. Öklid uzaklık formülünü internet üzerinden bulduktan sonra önceki işlemlerimizde frekans değerlerini kaydettiğimiz dizi ile bize verilen taslak kod içerisinde bulunan İngilizce ve Almanca frekans değerleri arasında Öklid uzaklığını hesaplayan kodumuzu yazdık. Ayrı Ayrı olarak bulduğumuz bu Öklid uzaklık değerlerini ekrana yazdırdık.

Beşinci ve son işlem olan metnin dilinin hesaplanması kısmına geçtik. Burada yaptığımız işlem kısa ve basitti. Çünkü uzaklık değeri küçük olan metnimizin dili o olacaktı. Bunun için yapmamız gereken koşul ifadeleri yazmaktı. Burada büyük bir problem ile karşılaştık. Çünkü bulduğumuz uzaklık değerleri eşit değildi ve bu sebeple sonucumuz doğru çıkmıyordu. Önceki işlemlerde hatalar yaptığımızı düşünüp tekrardan gözden geçirdik. Bu projeyi yapan diğer arkadaşlarımızda aynı sıkıntının onlarda da olduğunu fark ettik. Bu durumu hocalarımızla paylaştık. Aldığımız geri dönüt sonrası kodlarımızda sıkıntı olmadığını kabul ettik.

KAYNAKÇA:

1.Araştırma görevlisi hocalarımız

2.<https://www.yazilimgelistiricileri.com/c-karakterler-ve-karakter-dizileri/>

3.E kampüs Ders kayıtları