BLM209 PROGRAMLAMA LABORATUVARI I

PROJE 2

PROJE TESLİM TARİHİ: 19.11.2021

Projenin Amacı:

Bu projenin amacı sonek ağaçlarını ve sonek dizililerini kullanarak katarlar üzerinde bazı arama işlemleri yapmaktır.

Bir katar için sonek, katarın herhangi bir karakterinden başlayarak sonuna kadar olan kısımdır. Örnek olarak **bilişim** kelimesinin tüm sonekleri aşağıdaki gibidir.

1. bilişim (birinci karakterden başlayan sonek)
2. ilişim (ikinci karakterden başlayan sonek)
3. lişim (üçüncü karakterden başlayan sonek)
4. işim (dördüncü karakterden başlayan sonek)
5. şim (beşinci karakterden başlayan sonek)
6. im (altıncı karakterden başlayan sonek)
7. m(yedinci karakterden başlayan sonek)

Bir kelimenin öneki ise kelimenin ilk karakterinden başlayarak herhangi bir karakterine kadar olan kısımdır. Örnek olarak bilişim kelimesinin tüm önekleri aşağıdaki gibidir.

1. b (birinci karakterde sonlanan önek)
2. bi (ikinci karakterde sonlanan önek)
3. bil (üçüncü karakterde sonlanan önek)
4. bili (dörüncü karakterde sonlanan önek)
5. biliş(beşinci karakterde sonlanan önek)
6. bilişi(altıncı karakterde sonlanan önek)
7. bilişim(yedinci karakterde sonlanan önek)

Bir katarın (*p*) başka bir katar (*s*) içinde bulunması aslında *p*'nin *s*'in herhangi bir sonekinin öneki olmasını gerektirir. Örnek olarak *ili* katarı *bilişim* katarının içinde bulunup bulunmadığı bulmak için *bilişim* katarının tüm sonekleri oluşturulur ve *ili* katarının bu soneklerin herhangi birinin öneki olup olmadığına bakılır. Yukarıda listelenen soneklere bakıldığında *ili* katarı 2. sonekin önekidir ve *ili*

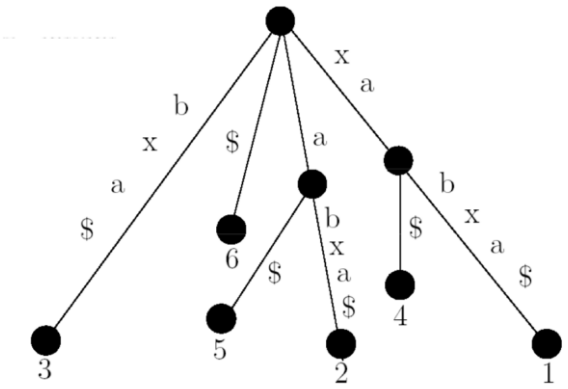
katarı *bilişim* katarının içinde yer alır. Aynı arama *ila* katarı için yapıldığında ise, *ila* katarı herhangi bir sonekin öneki olmadığı için *bilişim* katarı içinde yer almaz. Bu yaklaşımla bir katar içinde (uzunluğu *n* karakter olsun) başka bir kararı (uzunluğu *m* olsun) bulmak karmaşıklığı O(n+m).

Katarlar ne kadar uzun olursa olsun sınırlı sayıda farklı karakterin birleşmesiyle oluşur. Örnek olarak çok fonksiyonlu bazı proteinlerin uzunlukları binlerce karakter olabilirken bir protein dizilimleri 20 farklı karakterden oluşur. Böyle çok uzun katarlar içinde çok kısa birçok katarı aramak yukarıda anlatılan temel yöntemi kullanılarak yapılması pahalıdır. Ancak soneklere dayalı bazı veri yapıları kullanılarak arama işlemi çok daha ucuza (algoritmik karmaşıklık olarak) yapılabilir. Bu amaçla sonek ağaçları ve sonek dizileri geliştirilmiştir.

n uzunluklu *s* katarın sonek ağacı aşağıdaki özelliklere sahiptir:

- Ağacın *1*'den *n*'e kadar numaralandırılmış *n* adet yaprağı vardır.
- Kök dışında her düğümün en az iki çocuğu vardır.
- Her kenar *s*'in boş olmayan bir altkatarı ile etiketlenir.
- Aynı düğümden çıkan kenarların etiketleri farklı karakter ile başlamalıdır.
- Kökten başlayıp *k*. yaprağa giden yoldaki kenarların etiklerinin birleştirilmesi ile *k*. sonek elde edilir.

Örnek olarak *xabxa$* katarının sonek ağacı Figür 1'de verilmiştir.



Figür 1

Sonek ağaçları ve oluşturulmaları hakkında daha fazla bilgiye aşağıdaki linklerden erişilebilir.

- https://en.wikipedia.org/wiki/Suffix_tree
- https://web.stanford.edu/~mjkay/suffix_tree.pdf
- https://www.koseburak.net/blog/suffix-tree/

Bu proje kapsamında sonek ağaçları kullanılarak aşağıdaki problemler çözülecektir:

1. *s* katarı için sonek ağacı oluşturulabilir mi?
2. Sonek ağacı oluşturulan bir *s* katarı içinde *p* katarı geçiyor mu, geçiyorsa ilk bulunduğu yerin başlangıç pozisyonu nedir, kaç kez tekrar etmektedir?
3. Sonek ağacı oluşturulan bir *s* katarı içinde tekrar eden en uzun altkatar nedir, kaç kez tekrar etmektedir?
4. Sonek ağacı oluşturulan bir *s* katarı içinde en çok tekrar eden altkatar nedir, kaç kez tekrar etmektedir?

Yukarıdaki isterler grafiksel olarak gösterilmelidir. Grafik gösteriminin nasıl olacağı öğrenciye bağlıdır. Projede grafikler önemli bir yer tutmaktadır. Ne kadar özenli bir şekilde hazırlanırsa proje o kadar tamamlanmış sayılacaktır.

KISITLAR:

- Proje C dili kullanılarak geliştirilecektir.
- Kullanıcıdan alınacak parametreler (*s* katarının bulunduğu dosya, hangi problem çözülecek) komut satırından alınacaktır.
- *s* katarını barındıran dosya bir satırlık bir metin dosyasıdır.

ÖDEV TESLİMİ

- Ödevin raporu **LaTeX** kullanılarak yazılmalıdır.
- Rapor IEEE formatında (önceki yıllarda verilen formatta) 4 sayfa, akış diyagramı veya yalancı kod içeren, özet, giriş, yöntem, deneysel sonuçlar, sonuç ve kaynakça bölümünden oluşmalıdır.
- Dersin takibi projenin teslimi dahil edestek.kocaeli.edu.tr sistemi üzerinden yapılacaktır. edestek.kocaeli.edu.tr sitesinde belirtilen tarihten sonra getirilen projeler kabul edilmeyecektir.
- https://tr.overleaf.com/latex/templates/ieee-conference-template/grfzhhncsfqn adresinden IEEE formatlı örnek metin yapısını bulabilirsiniz.
- Proje ile ilgili sorular edestek.kocaeli.edu.tr sitesindeki forum üzerinden Arş.Gör. Emin Ölmez veya Arş.Gör. Yılmaz Dikilitaş'a sorulabilir.
- Sunum tarihleri daha sonra duyurulacaktır.
- Sunum sırasında algoritma, geliştirdiğiniz kodun çeşitli kısımlarının ne amaçla yazıldığı ve geliştirme ortamı hakkında sorular sorulabilir.
- Kullandığınız herhangi bir satır kodu açıklamanız istenebilir.

Sunum: Proje sunumu E-Destek üzerine yükleyeceğiniz projenizdeki kodlar indirilerek alınacaktır. Bu nedenle E- Destek üzerine yükleyeceğiniz projenin doğruluğundan emin olunuz.