

1 Özet

Bu sonек projesiyle ilk karşılaştığımızda bazı kısımların çözümlerini nasıl yapacağımızı tam olarak bilemedik. Sonек ağacını nasıl oluşturmamız gerekiyor linked listleri mi kullanmalıyız, isterlerdeki 1.soru için sonек ağacının oluşup oluşmama durumu nasıl belirleniyor bunu nasıl anlayacağız . 2.Soru için kullanıcının verdiği dizgeyi sonек ağacında mı aratmalıyız yoksa bir diziye atıp karakter karakter incelemeliyiz. Gibi problemlere çözüm başlangıçta bulamadık ama en büyüğüde sonек ağacının oluşturmaktan geçiyordu . Birde grafik olarak ne göstermeliyiz ve bu grafikleri nasıl oluşturacağımız konusunda bir düşünceye sahip değildik.Öncelikle linked listleri kullanarak bu sonек ağacını kendimiz oluşturmaya çalıştık lakin belli bir noktada hem harcadığımız zaman hemde aldığımız hata ve yaptığımız düzeltmelerden dolayı buna daha fazla zaman harcamayıp bir örnek araştırmaya başladık internette how can i create suffix tree diye araştırma yaptığımızda karşımıza çıkan çoğu şey farklı dillerde oluyordu. C’de bulunan açıklamaların ve kodların çoğu çok karmaşık ve kullanılabileceğimiz yapıda değillerdi. Son olarak bir sitede bir yazı bulduk bu yazı sayesinde hem sonек ağaçlarına hemde koda erişim sağlayabilmiş olduk ardından üzerinde değişiklikler yaparak istediğimiz formata çevirdik. Sonrasında görselleştirme için araştırmalarımızda eğitimlerler ve yazılar ile karşılaştık ama graphics.h kütüphanesi c programlama dilinde tam olarak çalışmıyordu. Bundan dolayı dil olarak C++ seçip c kodları ile devam etmek zorunda kaldık.

Nihayet projemiz tam anlamı ile bitti. yüzde 100 beklenen sonuca ulaşabilmekteyiz.

2 Giriş

Mevcut prolab 2 projesi olan sonек ağaçları projesinde temel anlamda istenilen structları , linked listleri, karmaşık bazı algoritmaları , okuma yazma işlemlerini kullanmamız . Bu işlemleri ise bize isterler kısmında çözülmesi gereken problemleri çözmemiz yönünde kullandırıyor sunuz.Bu problemleri şu şekilde açıklayabiliriz. Öncelikle kullanıcının girdi olarak vereceği şeyler ilk olarak metin belgesinin adı ve soru 2’de aranmasını istediği dizgi olacak onun haricinde çözülmesi gereken 4 farklı problem var bunların ilki textfile içerisindeki stringten bir sonек ağacı oluşturulabilir mi ve oluşturulabilirse farklı oluşturulamaz ise farklı bir grafik ve görselleştirme istenmekte Soru 2 için verilen dizge sonек ağacında bulunuyor mu bulunuyorsa farklı bulunmuyorsa farklı bir grafik istenmekte , soru 3 için en uzun alt dizginin gösterilmesini ve tekrar miktarının kontrol edilmesini isteniyor. Soru 4 için bakıldığında en çok tekrar eden sonек ağacındaki alt katarın tekrar miktarını ve kendisi istenmekte. Aynı zamanda Bu sonек ağacı projesinin birkaç farklı amacı mevcut. Bu amaçların başında projelerde yazdığımız kodlar sayesinde ileride kendi mesleklerimizde karşılaşacağımız problemlerin bir otomasyonunu yapar gibi yardımcı olmaktadır. Ve ilk sene öğrenmiş olduğumuz C programlama diline ait bilgileri unutmamızı engelleyip olabildiğince bizi kod yazma konusunda zinde tutuyorsunuz.

Bunun yanında farklı amaçları da var. Öğrenmiş olduğumuz dosya okuma yazma, string operatörleri ,linked list, graphics.h gibi bilgimizi kullanıp bu bilgilerimizi bir projede bir problemde çözüm olarak nasıl kullanmamız gerektiğini bize kendi kendimize çözüm bularak öğretmenizdir. Ve daha artırılabilir amaçları bulunmaktadır. Erişilen çıktılar analizlerin tam olarak söylenmesi gerekirse yüzde 100 üne doğru cevaplar vermektedir. Aynı zamanda projemiz isterlerin hepsini başarı ile gerçeklemektedir. fakat projemizin sıkıntısız devam ettiğinden emin olsak ta bazen teslerde problemler oluşabilmektedir.

3 Yalancı Kod

Metin belgesinin adını ad ;
Yapılacak seçimi al ;
Yapılan seçim 1 ise SonekMi fonksiyonuna dosyanın adını yolla ;
Adı gelen dosyayı oku ;
İçerisindeki bilgiyi katar isimli değişkene ata ;
Onek dizisini oluştur ;
Sonek dizisini oluştur ;
Onek ve sonek dizilerini karşılaştıır ve her eşitlik için sayacı artıık ;
Sayaç 3'ten büyükse ;
2 tane grafiğe 2 tane line ve " Boyle Bir Sonek Dizisi Yoktur !!!" yazdır ;
Sayaç 3'ten küçükse ;
X koordinatına 300 kordinatına 100 ata ;
Node sayısı kadar dönecek bir döngü başlat ;
Düğümün değeri -1 ise ;
Sonraki 2 düğümün verilerini 2 farklı değişkene ata ;
Sonraki 2 düğümünden sonraki 2 düğüm -1 'den sonraki 2 düğümle eşit mi diye kontrol et ;

Eşitlik varsa dur ;
Eşitlik yoksa eşitliğin olmadığı node'un verisinide farklı değişkene atayarak devam et;
Temp isimli değişkene Dallanma olan node'da kaçtane dallanmanın olduğu bilgisini dallanma numarası tut değişkeni ile ata ;
Düğüm sayısı kadar dönecek for döngüsü başlat ;
Düğümün değeri -1 ise temp değişkenini'de kullanarak OrtaAgaciCiz fonksiyonuna gönder ;
-1 'e eşit değilse direk o node'u OrtaAgaciCiz fonksiyonuna gönder ;
Yapılan seçim 2 ise belgenin adını pKatarı VarMi fonksiyonuna yolla ;
Adı gelen dosyayı oku ;
İçerisindeki bilgiyi katar isimli değişkene ata ;
Kullanıcıdan katarın al ;
Düğüm sayısı kadar dönecek bir döngü başlat ;
Düğümün değeri -1 mi diye bak -1 'e eşitse ;
O düğümün ilk karakterine eşitm i kontrol et ;
1 defa eşit olup olmadıysa onun alt düğümlerine eşitlik varmı kontrol et ;
Eşitlik varsa dallanmanın olduğu düğümü ve alt düğümü farklı değişkenlere at ;
Döngüyü break ile kır ;
Düğümün değeri -1 mi diye bak -1 'e Eşit değilse ;
O düğümün verisinin karakterleri stringin karakterlerine eşit mi bak ;
Eşit ise ;
O düğümün indexini farklı bir değişkene ata ve break ile dönügüy kır ;
X koordinatına 300 kordinatına 100 ata ;
Node sayısı kadar dönecek bir döngü başlat ;
Düğümün değeri -1 ise ;

Sonraki 2 düğümün verilerini 2 farklı değişkene ata ;
 Sonraki 2 düğümünden sonraki 2 düğüm -1 'den sonraki 2 düğümle eşit mi diye kontrol et ;
 Eşitlik varsa dur ;
 Eşitlik yoksa eşitliğin olmadığı node'un verisini farklı değişkene atayarak devam et;
 Temp isimli değişkene Dallanma olan node'da kaçtane dallanmanın olduğu bilgisini dallanma numarası tut değişkeni ile ata ;
 Düğüm sayısı kadar dönecek for döngüsü başlat ;
 Düğümün değeri -1 ise temp değişkenini'de kullanarak OrtaAgaciCiz fonksiyonuna gönder ;
 -1 'e eşit değilse direk o node'u OrtaAgaciCiz fonksiyonuna gönder ;
 Yapılan seçim 3 ise belgenin adını enUzunAltKatar fonksiyonuna yolla ;
 Adı gelen dosyayı oku ;
 İçerisindeki bilgiyi katar isimli değişkene ata ;
 Düğüm sayısı kadar dönecek bir for döngüsü kur ;
 Her düğümün verisinin uzunluğunu Enuzun değişkenine ata ;
 Düğüm sayısı kadar dönecek bir for döngüsü kur ;
 Enuzun değişkeninin değerlerini büyükten küçüğe sırala ;
 Enuzun değişkenin ilk indisindeki değere eşit olan düğümün numarasını dugumsayisi değişkenine ata ;
 X koordinatına 300 kordinatına 100 ata ;
 Node sayısı kadar dönecek bir döngü başlat ;
 Düğümün değeri -1 ise ;
 Sonraki 2 düğümün verilerini 2 farklı değişkene ata ;
 Sonraki 2 düğümünden sonraki 2 düğüm -1 'den sonraki 2 düğümle eşit mi diye kontrol et ;
 Eşitlik varsa dur ;

kontrol et ;
 Eşitlik varsa dur ;
 Eşitlik yoksa eşitliğin olmadığı node'un verisini farklı değişkene atayarak devam et;
 Temp isimli değişkene Dallanma olan node'da kaçtane dallanmanın olduğu bilgisini dallanma numarası tut değişkeni ile ata ;
 Düğüm sayısı kadar dönecek for döngüsü başlat ;
 Düğümün değeri -1 ise temp değişkenini'de kullanarak OrtaAgaciCiz fonksiyonuna gönder ;
 -1 'e eşit değilse direk o node'u OrtaAgaciCiz fonksiyonuna gönder ;

Yapılan seçim 4 ise belgenin adını enFazlaTekrar fonksiyonuna yolla ;
 Adı gelen dosyayı oku ;
 İçerisindeki bilgiyi katar isimli değişkene ata ;
 Düğüm sayısı kadar dönecek 2 adet for döngüsü kur ;
 Her nodu birbiriyle karşılaştır ve kaç adet olduğunu adetSay değişkenindeki değeri artırarak devam et ;
 Adetsay2 dizi oluştur ve adetSay 'ın değerlerini buna ata ;
 Düğüm sayısı kadar dönecek 2 adet for döngüsü kur ;
 Adetsay2 nin ilk indextindeki karakter adetSay değişkeninde hangi indiste indisi bul ;
 Bulunan indisi dugum sayısına ata ;
 X koordinatına 300 kordinatına 100 ata ;
 Node sayısı kadar dönecek bir döngü başlat ;
 Düğümün değeri -1 ise ;
 Sonraki 2 düğümün verilerini 2 farklı değişkene ata ;
 Sonraki 2 düğümünden sonraki 2 düğüm -1 'den sonraki 2 düğümle eşit mi diye kontrol et ;
 Eşitlik varsa dur ;

Eşitlik yoksa eşitliğin olmadığı node'un verisinden farklı değişkene atayarak devam et;
Temp isimli değişkene Dallanma olan node'da kaçane dallanmanın olduğu bilgisini dallanma numarası tut değişkeni ile ata ;
Düğüm sayısı kadar dönecek for döngüsü başlat ;
Düğümün değeri -1 ise temp değişkenini'de kullanarak OrtaAgaciCiz fonksiyonuna gönder ;
-1 'e eşit değilse direk o node'u OrtaAgaciCiz fonksiyonuna gönder ;

4 Temel Bilgiler

Bu sonек ağacı projesinin kodlarını yazarken dev c++, graphics kütüphanesi ve internetten bulduğumuz çeşitli bilgilerden yararlandık

5 Yöntem

Soneк ağacı projesinin başlangıcında kullanıcıdan girilmesi gereken text file'ın adını alıyoruz ardından bu isimi yapılan seçime göre istenilen fonksiyona gönderiyoruz. İsterlerdeki ilk soru'da main'den gelen dosyanın ismi açılıyor ve içerisindeki bilgi katar isimli değişkene alınıyor ardından bu değişkenin sonек ve önek dizilerini oluşturuyoruz . Soneк ağaçları projesinde sizin kaynak olarak bakabilirsiniz dediğiniz yazıların birisinde geçen bilgiden yararlanarak bu soruyu çözmüş bulunmaktayız. Bu bilgi ise sonек dizileri ve önek dizilerinde ortak karakter bulunamaz.Eğer bulunuyorsa o stringten sonек ağacı oluşmaz deniyordu. Bizde oluşturmuş

olduğumuz sonек ağaçlarını ve önek ağaçlarını strcmp fonksiyonunu kullanarak karşılaştırıp ortak alt dizgileri varmı diye kontrol ettik eğer mevcut ise sonек ağacı oluşturulamaz . Mevcut değilse o zaman sonек ağacını oluşturup görselleştirme işlemlerini yaptık görselleştirme işlemlerini yaparken aslında sürekli oluşturulan sonек ağacının düğümlerindeki bilgileri kullanıyoruz. Suffix tree'yi oluşturduktan sonra düğüm sayısı belli olduğundan o kadar dönecek bir dönüyle o düğümde parçalanma olup olmadığını kontrol ederek ve koordinatları da gerekli şekilde değiştirerek ortaAgaciCiz fonksiyonuna yollayıp gerekli daire ve line'ları bastırıyoruz. 2.sorunun çözüm kısmında Soneк ağacını oluşturuyoruz. Oluşturulan sonек ağacının düğüm sayısı her düğümündeki veriler belli olduğundan bunlarla kullanıcının girdiği stringin karakter karakter kontrol ediyoruz. Tüm karakterler eşit olduğunda bir break ile döngüyü kırıyoruz. Ve döngü kırılmadan önce o anki düğümün indexini bir değişkene atıyoruz görselleştirme için OrtaagaciCiz fonksiyonuna gittiğinde bu kontrolden geçip ona göre o node'u farklı renkte çiziyor Soru 3 te öncelikle tüm düğümlerin uzunluklarını strlen ile alıyoruz ve bir değişkene atıyoruz. Bu değişken sayesinde artık tüm düğümlerdeki verilerin uzunluklarına sahibiz sonrasında bu uzunların arasında en büyük olan değeri buluyoruz ve onun bulunduğu indisteki düğümün indexini bir değişkene atıp görselleştirme işlemlerini gerçekleştiriyoruz. Diğer kısımda olduğu gibi görselleştirme işlemleri gerçekleştirirken değişkenin değeri değiştiğinden o düğüm farklı renkte basılıyor . Soru 4 için aynı soru

3 yapmış olduğumuz gibi öncelikle for içinde for sayesinde tüm düğümlerdeki verilen kaç defa döndüğünü buluyor ve bir integer dizisinde saklıyoruz. Sonrasında bu dizinin en büyük elemanının bulunduğu indisteki düğümün rengini görselleştirme kısmında farklı renkte bastırıyoruz.

6 Sonuçlar

Sonek ağaçları projesine bugüne kadar yaptığımız projelerde daha fazla süre ayırdık ve daha fazla çaba harcadık. Bu emek ve çabalar eşliğinde programımız yüzde 100 oranında doğru çalışmaktadır. yine de proje başlangıcında eksik olan yanlarımız yok denecek kadar azdı. internetten araştırmalar yapıp bunları öğrendik ve koda geçirdik şuanda projemizin çalışma oranında yüksektir. Ve de herhangi bir sıkıntı veya bir sorun bulunmamaktadır.

7 Kaynakça

<https://www.geeksforgeeks.org/representtreeusing-graphics-in-c-c/>
<https://en.wikipedia.org/wiki/Suffixtree>
<https://qastack.info.tr/programming/9452701/ukkonenssuffixtree>
<https://www.geeksforgeeks.org/generalizedsuffixtree-1/>
<https://hknsoft.com/cprogramlamadizideki-sayilari-buyuktenkucugesiralama/>
<https://www.koseburak.net/blog/suffixtree/>
https://www.tutorialspoint.com/data_structures_algorithms/link
<https://stringfixer.com/tr/Suffixtree>
<https://web.stanford.edu/mjkay/suffixtree.pdalgorithm-in-pla-in-english>
<https://www.programiz.com/c-programming/c-structures>
<https://www.geeksforgeeks.org/structures-c/>
<https://www.programiz.com/c-programming/library-function/string.h/strcmp>