Kocaeli Üniversitesi

Bilgisayar Mühendisliği Bölümü

Programlama Laboratuvarı-1

Mahmut Emre Terzi-180 201087
Mahmut Bilgi-200201 037

Özet

Bu dokümanda Programlama Laboratuvarı1 dersi 2. Projesi için çözümümü açıklamaya yönelik oluşturulmuştur.Dokümanda projenin tanıtımı, çözüme yönelik yapılan araştırmalar, kullanılan yöntemler, proje hazırlanırken kullanılan geliştirme ortamıve kod bilgisi gibi programın oluşumunu açıklayan başlıklara yer verilmiştir. Dokümanın sonunda projemizi hazırlarken kullandığımız kaynaklar ve projederlenirken dikkat edilmesi gereken hususlar bulunmaktadır.

1-Proje Tanıtımı

Bu projenin amacı sonek ağaçlarını ve sonek dizilerini kullanarak katarlar üzerinde bazı arama işlemleri yapmaktır.
Bir katar için sonek, katarınherhangi bir karakterinden başlayaraks onunakadar olan kısımdır.Örnek olarak bilişim kelimesinin tüm sonekleri aşağıdaki gibidir.
1. bilişim(birincikarakterden başlayan sonek) 2. ilişim(ikinci karakterden başlayansonek)
3. lişim(üçüncü karakterden başlayan sonek)
4. işim(dördüncü karakterden başlayan sonek) 5. şim (beşincikarakterden başlayansonek)
6. im (altıncıkarakterden başlayan sonek)
7. m(yedinci karakterden başlayan sonek)

Bir kelimenin önekiise kelimenin ilk karakterinden başlayarakherhangi bir karakterine kadar olan kısımdır. Örnek olarak bilişimkelimesinin tüm önekleri aşağıdakigibidir.
1. b (birinci karakterde sonlanan önek)
2. bi (ikinckarakterde sonlanan önek)
3. bil (üçüncü karakterde sonlanan önek)
4. bili (dördüncü karakterde sonlanan önek)
5. biliş(beşinci karakterde sonlanan önek)
6. bilişi(altıncıkarakterde sonlanan önek)
7. bilişim(yedinci karakterde sonlanan önek)

Bir katarın(p) başka bir katar (s) içinde bulunmasıaslında p'nin s'in herhangi bir sonekinin önekiolmasını gerektirir. Örnek olarak ilikatarı bilişim katarının içinde bulunup bulunmadığı bulmak için bilişim katarınıntüm sonekleri oluşturulur ve ilikatarının bu soneklerin herhangi birinin önekiolup olmadığına bakılır. Yukarıdalistelenen son eklere bakıldığındaili katarı2. sonekin ön ekidir ve ilikatarı bilişim katarınıniçinde yer alır. Aynıarama ilakatarıiçin yapıldığındaise, ilakatarıherhangi bir sonekin önekiolmadığı için bilişim katarı içinde yer almaz. Bu yaklaşımla bir katar içinde (uzunluğun karakter olsun) başka bir kararı (uzunluğum olsun) bulmak karmaşıklığı O(n+m).

Katarlar ne kadar uzun olu rsa olsun sınırlı sayıda farklıkarakterin birleşmesiyle oluşur. Ö rnek olarak çok fonksiyon lu bazı proteinl erin uzunluklarıbinle rce karakter olab ilirken bir protein dizilimleri 20 farklıkarakterden oluşur. Böyle çok uzun katarlar içinde çok kısa birço k katarıaramak yukarıda anlatılantemel yöntemi kullanılarak yapılması paha lıdır. Anca k so neklere dayalı bazı veri yapıları kullanılarakara ma işlemi çok daha ucuza (algori tmik karmaşıklık olara k) yapılabilir. Bu ama çla sonek ağaçl arıve sonek dizile ri geliştirilmiştir.

n uzunluklu s katarınsonek ağa cı aşağıda kiözellikl ere sahiptir:

- Ağacın 1'd en n'e kada r numara landırılmışn adet yaprağı vardır.
- Kökdüşünda her düğümün en az iki çocuğu vardır.
- Her kenar s'in boş olmaya n bir altkatarıile e tiketlenir.
- Aynı düğümd en çıkan kenarla rın etiketleri farklı karakter ile başla malıdır.
- Kökten başl ayıp k. yaprağa gide n yolda ki kenarların etikl erinin birleştirilmesiile k. son ek elde edil ir.

Örnek olara k xabxa$ katarınınsonek ağacı Figür 1' de verilmiştir

2-Proje yöntemi Deney ve Araştırma

Bu proje kapsamında sone k ağaçları kullanılarak aşağıda kiprobleml er çözülme si istenmektedir:

1. s katarıiçin sonek ağ acı oluşturulabilir mi?
2. Sonek ağa cı oluşturulanbi r s katarı içinde p katarıgeçiyor mu, geçiyorsa ilk bulunduğu yeri n başlang ıçpo zisyonu nedi r, kaç kez tekrar etmektedir?
3. Sonek ağa cı oluşturulanbi r s katarı içinde te krar eden en uzu n altkatar nedir, kaç kez tekrar etmektedir?
4. Sonek ağa cı oluşturulanbi r s katarı içinde en çok tekrar eden al tkatar nedir, kaç kez tekrar etmektedir? Yukarıdaki isterler grafiksel olarak göste rilmek istenmekted ir. Öncelikli olarak projede veri yapıları ve ağa ç yapısını öğ renmemiz gerekl iydi. Ağaç yapısı kısaca şöyle anlatılmaktadır. Ağaç (Tree) ve ri yapısı çok yaygın ola rak kullanılançok güçlü bir veri yapısıdır. Ağaçlar (Trees) doğ rusal (line er) veri yapıları olan diziler, bağlantılılisteler, yığınlarve kuyrukl ardan farklıolara k, doğru sal olmayan hiyerarşik bir veri yapısıdır. G ündelik hayattan bildiğimizsoy ağacı ağa ç ve ri yapısı hakkında bize fikir verebi lir. Ağaç, verilerin birbirine s anki bir ağaç yapısı oluşturuyormuş gibi sa nal olara k bağlanma sıylaelde edi lir. Buyapıda veri, düğümlerde (nod e) tutulur. Düğümle re ağcın ele manı denir. Ağaç veri yapısında düğümler arası ilişkiken arlar /dallar (edge s) kullanılarak oluşturulur. Başka bi r ifade ile ağaç veri yapısında düğümler birbirine kenarl ar/dallar kullanılarak bağlanır. Aşağıda ağaç yapısı gösteril miştir.

Ağaç yapısında nsonra proje deki asıl ister olan suffix tree modellemesi ni öğren memiz gerekti.
Suffix Tree'nin Tanımı:
m uzunluğundakibir S string için T suffix tree aşağıdaki özellikle re sahip tir:
• Köklü bir ağaçtırve yönlüdür
•1 ilem arasında etiketlenmiş m yaprağı vardır
• Ağaçtakihe r bir dal S string nin bir alt stringini oluşturur
• Kökten, i.yaprağa kadar etiketlenmişbir yol üzerin deki kenarlar birleştirilebilir• Kök olmayan her ara düğümün en az 2 yaprağı vardır
• Birdüğümden çıkan kenarla r farklı karakterle r ile başlar.
S=abab
S string'inin suffix tree'si, S'nin bütün suffix'lerini sıkıştırılmışbir trie de tutsun. $ sembolü ilgili suffix'in sonun u göstersin.
{ $, b$, ab$, bab$, aba b$ }

Suffix ağacımız yukarıdaki şe kilde oluşturulmuştur.Suffix ağaçlarının işe yaradığı alanları şöyle anl atabiliriz.
Suffix Tree (Sonek Ağacı) keli me işleme algoritmalarındandır.
DNA dosyaları gigabyte se viyesinde yer kapladıklarındanDN A analizinin elle yapılması mümkün değildir.
Hatta, DNAdosya larınınbilgi sayar yardımıyla işlenmesi de çok uzun sürmektedi r.
Biyolojik veriler,ara ma motorları, derle yici tasarımı, işletimsistemi, veri tabanı,vs.. kullanılır.

3-Yala ncı Kod

```
include
Struct n oluştur;
Struct n altındaStruct n*çocuk[n] oluştur;
Struct n altındaint basla ,int son ,int index
oluştur;
Char katar[N]boyutlu dizisini tanımla;
İntdugumMu fonksi yonunu tanımla;
İntdugumMu fonksi yonuna x,y,z
parame trelerini ata;
İntdugumMu fonksi yonunda x==z ye ise return
x-1 döndür;
İntdugumMu fonski yonunda katar[x]==katar[y]
ise
Return dugumMu fonksiyon un içinden
x=x+1,y=+1,z=z döndür;
İntdugumMu fonski yonunda isten enler
sağla nmıyor(else) ise return x-1 döndür;
void dalYap fonksiyon unu tanımla;
İntdalYap fonksi yonunda parametre olarak *n
kullan;
void dalYap fonksiyon unda katar karakter dizisi
boyutuna eş int M tanımla;
void dalYap fonksiyon unda M kadar dön gü
tanımla;
void dalYap fonksiyon unda çocuk,son ve in dexi
kullanarak işlemleriyap;
void icDugum tanımla,parametre olarak
node *root,int i kullan;
void icDugum içinde int t=0,int m=katar karakter
dizisiboyutu,int bitiş[10],int başlang ıç[10 ],int
ilk[10] tanımla;
void icDugum içinde 3 farklıdöngü oluştur;
void icDugum içinde ikisi j ve k ya kadar olucak
şekilde biri ise eşit değildiri ye ve k M den
küçükse şeklindedöng üler oluşturarakicDugum
sağla nmış olucaktır;
void icDugum2 tanımla,parametre olarak
node *root kullan;
void icDugum2 için de int M=katar karakter dizisi
boyutu,int bitiş[10],int başlangıç[10],int ilk[10]
tanımla;
void icDugum2 için de 3 farklıdöngü oluştur;
void icDugum2 için de biri M ye kadar biri t ye
kadar diğerik i ye eşitolmayaca k ve k M den
küçük olana kadar olucak şeklinde döngüler
oluşturularakicDug um2 sağlanmış olucaktır;
void Dugumici tanımla,parametre olarak
node *root kullan;
void Dugumiçi içinde katar karakter dizisi
boyutunda int Mtanımla;
void Dugumiçi içinde dalyap fonkisyonu içine
rootu gönder ;
void Dugumiçi içinde M ye kadar oluca k döngü
şeklin deDugumi çini oluştur;
```

void Dugumici2 tanımla,parametre olarak node *root, node*n tanımla;
void Dug umici2 içinde katar karakter dizisi boyutunda int M değişken i tanımla;
void Dugumici2 içinde b tanımla;
void Dugumici2 içinde dalya p fonksiyonu içinde rootu gönde r;
void Dugumici2 içinde M ye kadar olu cak şekilde Dugümiçi 2 yi oluştur;
void grafik fonksiyonunu tanımla,parametre olarak node *n kullan;
void grafik fonkisyonu içinde ağaç gösterimi katar gösterimi graphic h kütüphanesinde ki işlevlerikullan arak gösterme ve çizdirm e döng ülerini ayarlamala rını yap;
void sorg ula fonskiyon unu tanımla,parame tre kullanma;
void sorg ula fonksiyon unda N boyutunda string oluştur,system(cl s) komutunu tanımla;
void sorg ula fonksiyon unda string girişi ve bu string girişiniint A ya atama yap;
void sorg ula fonksiyon unda A için onek ve sonek cha r formatında tanımlamalar yap;
void sorg ula fonksiyon unda sonek on ek oluşu mlarına bakara k katar için sonek ağacı oluşturulabilir değilse oluşturulumaz çıktılarını oluştur;
void olustur fonskiyonun u tanımla ve parame tre kullanma;
void olustur fonkisyonun un içinde str [N] boyutunda char tanımla,string girişi için tanımlama yap ;
void olustur içinde dalyap fonksiyon una root gönd er;
void oluşturiçinde çocu k oluşturdöngü ile;
void oluşturiçinde icDugumu kullan ve içine root ve i gönder;
void oluşturiçinde icDugumude n sonra dugu mici2 ve icdug um2 fonskiyonla rını döng ü içinde M ye kada r olucak şekilde oluştur;
void oluşturiçinde tekrar ed en katarı,en uzun alt katarı ve tekrar eden al t katarı bul mak için, ;
void oluşturiçinde int tut,durum,basla,so n,durum,tekrar=0 tanımlamalarını yap;
void oluşturiçinde katar arama işlevleri ni yap;
void oluşturiçinde suffix tree bulunma bulunmama durumlarını gö ster;
main fonksi yonu tanımla,while döngüsü ile menü oluştur;
main içinde 2 seçim tanımla,ilseçim sorgula fonksiyon u,ikinci seçim oluşturf onskiyonunu çağır,remove il e katar texti yaz ve kapat;
return0;

4-Kayna kça

https://www.kosebura k.net/blog/suffix-tree/

https://www.it-swarm-tr.com/tr/suffix-tree/

https://cdn-acikog retim.istanbul.ed u.tr/auzef content/20_21_Guz/veri_yapilari/9/index.html

https://avesis.yildiz.ed u.tr/resume/downloadfile/diri?key=7aa1aefa-a34 2-4dca-ba 27-27e3 8b5359 ec

https://qastack.info.tr/programming/9452701/ukkone ns-suffix-tree-algorithm-in-plain-eng lish

https://nerdbook.wo rdpress.com/2018/03/28/agac-veri-ya pisi/

https://cdn-acikog retim.istanbul.ed u.tr/auzef content/20_21_Guz/veri_yapilari/9/index.html

https://www.youtube.com/wa tch?v=VEkAj-xVTKQ&t=187s