

# KOCAELI UNİVERSİTESİ

## BİLGİSAYAR MÜHENDİSLİĞİ

-

### PROLAB I-2

### SUFFIX TREE

Muhammed Numan MERCAN  
200202098

Mehmet Ali AKDOĞAN  
20202017

#### I. BAŞLARKEN

Bu projede sonek ağaçlarını (suffix tree) kullanarak stringler üzerinde arama işlemleri yapılması istenmiştir.

Suffix, bir stringin bir karakterinden sonuna kadar olan her karakter dizisine verilen isimdir. Bir katarın (p) başka bir katar (s) içinde bulunması aslında p'nin s'in herhangi bir sonekinin öneki olmasını gerektirir.

Sonek ağacını daha iyi anlamamız için şöyle bir tanım yapabiliriz:

n uzunluklu s stringin suffix tree si su özelliklere sahip olur :

- 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 etiketlerinin birleştirilmesi ile k. sonek elde edilir.

#### II. ÖZET

Projeye suffix algoritmasını verilen kaynaklardan mantığı anlamaya çalışarak başladık. Karmaşık ve istenilen dilde örneği çok kısıtlı olan bu algoritmayı mantık olarak öğrendikten sonra kısım kısım kodlamaya çalışarak C dilinde pratiğini yaptık, fakat kodlama kısmi beklediğimizden daha zor gelişti ve

ilerlemek için çok vakit harcamamız gerekti.

Bu cabaların ardından 2. Büyük ister olan bu algoritmayı grafiksel bir arayüze taslamak için grafik kutuphanesi ve kodlarını öğrenmeye koyulduk. Yine uzun süren bir cabanın ardından yavaş yavaş göze hitap edebilen bir arayüz geliştirdik. Fakat bu arayüzde normal terminalde çalışan kodlar gibi bir kolaylık söz konusu değildi çünkü kodlar otomatik bir şekilde değil manuel olarak verilmesi gereken koordinatlara göre yerleşiyorlardı. Uzun süren uğraşlar sonunda tam olarak bitmemiş olsada iyi bir yol kat edilmiş proje elimizdeydi.

Bu raporda da yaptığımız bu projeyi her tarafıyla paylaşıyor olacağız.

#### III. GİRİŞ

Sonek ve onek ilişkisi ile algoritmayı oluşturmaya başladık, basta ilkel yöntemlerle normal karakter dizileriyle kontrol ve etkileşimleri oluşturmaya çalıştık. Ardından daha kompleks şekilde istekleri tamamlamaya başladık, ağaç yapısını kurduk ve arama işlemleri üzerinde uğraştık. Bu kısımda yapabildiğimizi yaptık. Bu işlemler yeterli seviyeye geldiğinde yavaş yavaş grafiksel arayüze aktarmaya başladık. Önce koordinatları belirleyerek terminaldeki gibi bir string listesi ortaya çıkarttık. Ardından bu dizgilerin ağacın hangi yaprakları olduğunu biraz karmaşık bir sıralama ile göstermeyi basardık. Tabii normal bir ağaç yapısı yakalamamız pek kolay olmadığı için kısıtlı süre içinde yapabileceğimiz bir grafik oluşturduk.

#### A. YONTEM ve İLERLEYİŞ

Biz öncelikle bir ağaç yapısını kolayca kurup kullanmayı ardından onu grafik arayüzüne nasıl aktarabileceğimizi öğrenmeye çalıştık, tam olarak bir planımız yoktu çünkü suffix tree bizimde ilk defa karşılaştığımız bir kavramdı bu yüzden akış seması nı oluşturmadan projeye başladık. Pek verimli olmasa da bir ilerleme kaydetmemizi sağladı bu yöntem.

Amacımız ağaca benzer en yakın ve anlaşılır grafiği elde etmek olduğu için öncelikle örnek benzer ağaç yapılarını ve onları terminale nasıl çizildiğini inceledik. Bu bize işe yarar fikirler verdi ve o şekilde ilerlemeye karar verdik.

#### B. Hedeflenen Örnekler

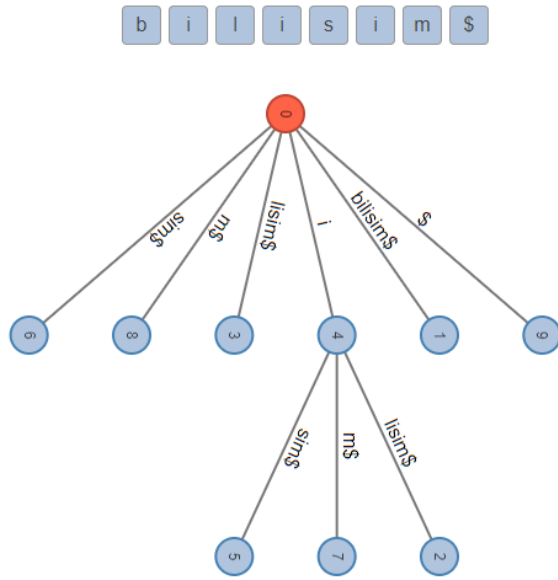


Fig. 1. Örnek bir suffix tree

Bu ağaç yapısı tam anlamıyla bir suffix treeye karşılık gelir. Suffix tree mantığını dallanmalardan anlayabilirsiniz. Aynı karakterle başlayan sözcükler ortak bir dallanmayla dağılıyorlar, bazı işlemleri bu dallanma hesapları ile yapıyor olacağız.

Aşağıda Suffix tree'yi grafiksel olarak ve terminalde semboller ile gösterimini bulacaksınız

#### IV. SONUÇLAR

Bizim projede gelebildiğimiz nokta isteklerin çoğunu karşılamakla birlikte uzun uğraşlar verildiği halde çözilemeyen bazı kısımların olduğudur.

```
C:\Users\numan\Desktop\Tree\SuffixTre\bin\Debug\SuffixTre.exe
+
+-- bilisim$
++ i
| +-- lisim$
| +-- sim$
| +-- m$
+-- lisim$
+-- sim$
+-- m$
+-- $

Process returned 0 (0x0)   execution time : 0.094 s
Press any key to continue.
```

Fig. 2. Terminalde yapılmış bir ağaç örneği

Bu örnekte C programlama ile yapabileceğimiz grafiğe daha yakın bir örnek sunuyor. Dallanmalar örnekte de anlaşılacağı üzere daha ucuz ve akıllıca bir biçimde bir kaç adet sembol ve boşluk ile halledilmiş durumda.

Basarılı kısımları ve hatalarımızı raporun ilerleyen kısımlarında paylaşıyor olacağız. Bu projenin sonunda suffix tree'yi prefix'leriyle birlikte araştırmanın daha kolay olduğunu öğrendik. Suffix tree genelde biyolojik bilimsel araştırmalarda kullanılmakla birlikte bazı kolaylıklar sağladığı farklı alanlarda bulunuyor, biz de çok örneğe de bu tarz alanlarda araştırarak ulaştık

Ana Semaya sigmayan fonksiyon semalarını ayrı semalar olarak paylaştık.

```
Lutfen stringi giriniz: bilisim$
Girilen string: bilisim$
-$ [7]
-bilisim$ [0]
-i [-1]
-lisim$ [1]
-m$ [5]
-sim$ [3]
-lisim$ [2]
-m$ [6]
-sim$ [4]
```

Fig. 3. Terminaldeki demomuz

Bu bizim uykunnen algoritmasını araştırırken öğrendiğimiz ve kendimize uyarladığımız terminalde yakaladığımız referans sonuç çıktımız.

```

+ $      [7].yaprak
+
+ bilisim$[0].yaprak
+
+ i      [-1].yaprak
+
+ lisim$ [1].yaprak
+
+ m$     [5].yaprak
+
+ sim$   [3].yaprak
+
+ lisim$ [2].yaprak
+
+ m$     [6].yaprak
+
+ sim$

```

Fig. 4. Grafik ciktisi demomuz

Ayni algoritma ve terminal ciktisindan esinlenerek koordinatlari ayarlayip olusturdugumuz grafik arayuzumuz. Bu kisim demo kisim ilerleme kaydedilen hali de paylasilacaktır. Bu paylasimlarimiz ve esinlendigimiz tum kaynaklari da asagida kaynakca kisiminda paylasiyor olacagiz.

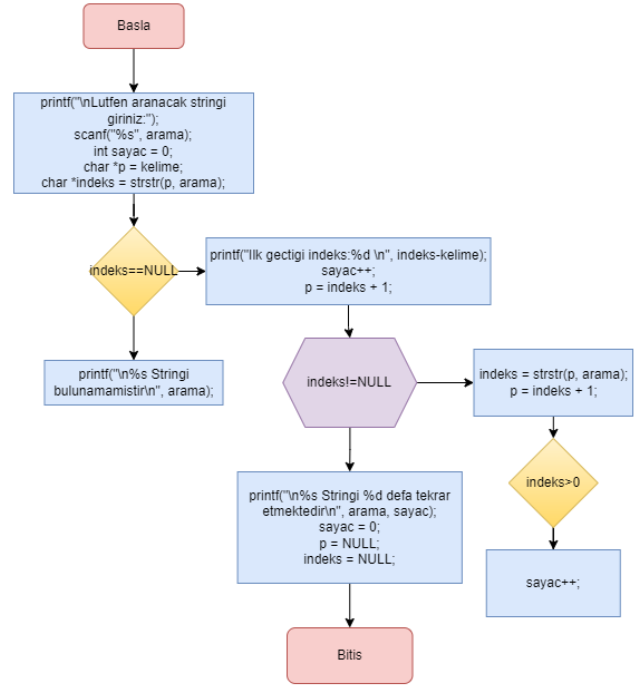


Fig. 6. Akis Semasi case 3 func. String bulma FlowChart ,

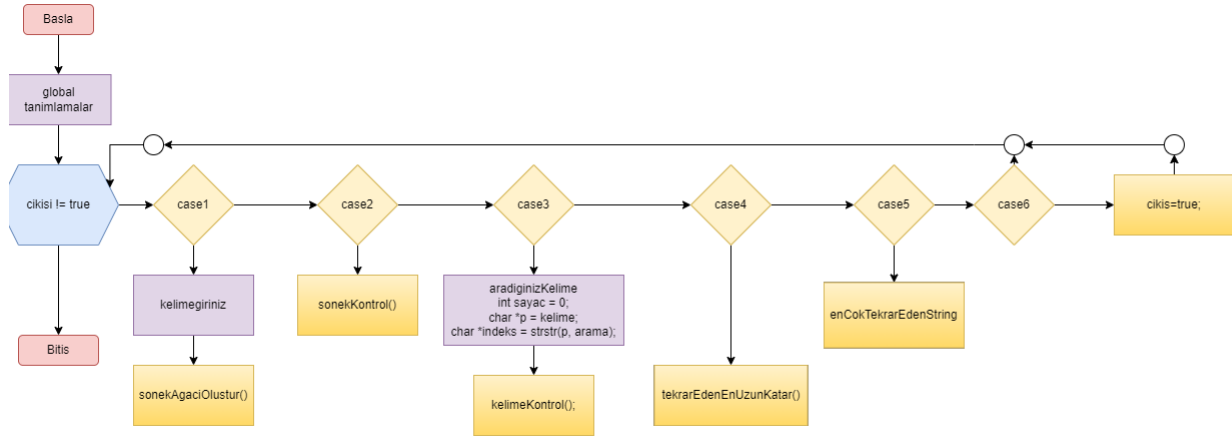


Fig. 5. Akis Semasi Ana Govde

Main FlowChart , basic links

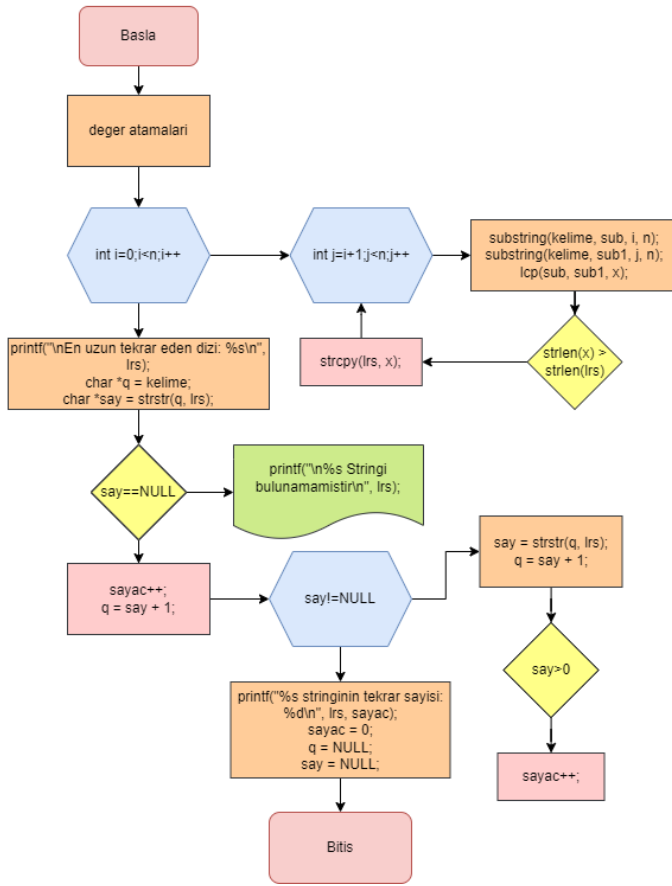


Fig. 7. Akis semasi case 4 func

## V. KAYNAKÇA

- <https://en.wikipedia.org/wiki/Suffix-tree>
- <https://web.stanford.edu/~mjkay/suffix-tree.pdf>
- <https://stackoverflow.com/questions/9452701/ukkonens-suffix-tree-algorithm-in-plain-english/95134239513423/>
- <https://www.bigocheatsheet.com/>
- <https://github.com/brenden/ukkonen-animation>