



Yıldız Teknik Üniversitesi
Elektrik-Elektronik Fakültesi
Bilgisayar Mühendisliği Bölümü

BLM2512 Veri Yapıları ve Algoritmalar
Ödev – 3

Öğrenci Adı: Ali Mert Temizsoy

Öğrenci Numarası: 23011018

Dersin Eğitmeni: Mehmet Amaç Güvensan

Video Linki: <https://youtu.be/-yV9rUf80A0>

1-Problemin Çözümü:

Bu ödevde şirket hiyerarşisini ağaç veri yapısı kullanarak modellemek ve bu yapı üzerinde çeşitli analizler yapmak amaçlanmıştır. Şirket hiyerarşisi, en üstte CEO'nun bulunduğu ve her seviyede farklı unvanlara sahip çalışanların yer aldığı bir ağaç yapısı ile temsil edilmektedir.

Çözüm için tasarlanan algoritma ve veri yapıları aşağıdaki gibidir:

1. Öncelikle çalışan bilgilerini saklamak için bir **Employee** struct yapısı tasarlanmıştır. Bu yapıda çalışanın adı, yaşı, maaşı, çocuk sayısı, ebeveyn bilgisi ve çocuklarını tutan bir işaretçi dizisi bulunmaktadır. Bu struct, ağaç veri yapısının düğümlerini temsil etmekte ve çalışanlar arasındaki hiyerarşik ilişkiyi modellemektedir.
2. Algoritma ilk olarak bir dosyadan çalışan bilgilerini (ad, yaş, maaş ve yönetici bilgisi) okur. Bu bilgiler iki geçici dizide tutulur: Biri çalışan nesnelerini, diğeri ise her çalışanın yönetici adını saklar. Bu aşamada henüz ağaç yapısı oluşturulmamıştır; çalışanlar arasında hiyerarşik ilişki kurulmamıştır.
3. Geçici dizilerdeki bilgiler kullanılarak ağaç yapısı oluşturulur. Bu süreçte, her çalışan için yönetici bilgisi kontrol edilir:
 - Eğer bir çalışanın yöneticisi "NULL" ise, bu çalışan CEO'dur (ağaç yapısının kökü).
 - Diğer çalışanlar için, yönetici adı geçici dizide aranır ve bulunduğunda, çalışan ile yönetici arasında parent-child ilişkisi kurulur.
 - Bu işlem, çalışanın parent işaretçisinin yöneticiye, yöneticinin children dizisine ise çalışanın eklenmesiyle gerçekleştirilir.
 - Böylece, düz veri yapısından hiyerarşik bir ağaç yapısı oluşturulmuş olur.
4. Oluşturulan ağaç yapısı üzerinde çeşitli analizler rekürsif algoritmalar kullanılarak gerçekleştirilir: Tüm hesaplamalarda temel yaklaşım benzerdir: Rekürsif fonksiyonlar ağaç yapısında dolaşarak, her düğüm için belirli işlemleri gerçekleştirir ve sonuçları yukarıya doğru birleştirir. Her düğüm için yapılan işlem, o düğümün tüm çocuklarına rekürsif olarak uygulanır ve elde edilen sonuçlar birleştirilerek düğümün kendi sonucu oluşturulur. Örneğin, ağacın yüksekliği hesaplanırken, her düğüm için tüm çocukların yükseklikleri rekürsif olarak hesaplanır, bu değerlerden en büyüğü alınır ve 1 eklenerek düğümün kendi yüksekliği bulunur. Seviyedeki personel sayısı hesaplanırken, istenilen seviyeye ulaşana kadar rekürsif çağrılar yapılır ve ulaşıldığında sayaç artırılır. Yaş ortalaması için toplam yaş ve çalışan sayısı eş zamanlı olarak hesaplanır; her düğümün yaşı toplama eklenir ve çalışan sayacı bir artırılır, rekürsif çağrılar tamamlandığında toplam yaş çalışan sayısına bölünerek ortalama bulunur. Bu rekürsif yaklaşım, ağaç veri yapısında verimli bir şekilde dolaşmayı ve gerekli bilgileri tek bir geçişte toplamayı sağlar.
5. Son olarak, hesaplama sonuçları kullanıcıya gösterilir ve dinamik olarak ayrılan bellek alanları serbest bırakılır.

Bu algoritmada, ağaç veri yapısının özellikleri ve rekürsif fonksiyonların gücü kullanılarak, karmaşık bir hiyerarşik yapı etkin bir şekilde modellenmiş ve analiz edilmiştir. Rekürsif yaklaşım, özellikle ağaç yapılarında derinlik ve genişlik hesaplamaları için ideal bir yöntemdir.

2-Karşılaşılan Sorunlar:

Algoritmanın tasarımı ve implementasyonu sırasında çeşitli zorluklarla karşılaşılmıştır:

Ağaç Yapısının Doğrudan Oluşturulma Problemi: İlk implementasyonda, dosyadan okunan verileri doğrudan ağaç yapısına dönüştürmeye çalıştım. Bu yaklaşımda, her çalışan bilgisi okunduğunda, mevcut ağaç yapısında o çalışanın yöneticisini aramak ve bağlantıyı kurmak gerekiyordu. Ancak, henüz ağaç yapısı tam oluşmadığı için, bazı çalışanların yöneticileri ağaçta bulunamıyor ve bağlantılar eksik kalıyordu. Bu sorunu çözmek için, iki aşamalı bir yaklaşım benimsedim: İlk aşamada tüm çalışan bilgilerini ve yönetici bilgilerini geçici dizilerde sakladım, ikinci aşamada ise bu geçici dizileri kullanarak tam ağaç yapısını oluşturdum. Bu sayede, her çalışanın yöneticisi kolayca bulunabiliyor ve hiyerarşik ilişkiler doğru şekilde kurulabiliyordu.

Bellek Tahsisi ve İşaretçi Yapısı Sorunları: Çalışan bilgilerini tutmak için kullanılan employeeTemp dizisini başlangıçta tek bir işaretçi (Employee*) olarak tanımladım. Bu yaklaşım, düzgün bellek tahsisi yapmayı zorlaştırdı ve erişim hatalarına neden oldu. Sorun, employeeTemp dizisini çift işaretçi (Employee**) olarak yeniden tanımlayarak çözüldü. Bu sayede, her çalışan için ayrı bir Employee yapısı oluşturulabildi ve bu yapılara doğru şekilde erişilebildi. Ayrıca, çift işaretçi kullanımı, diğer işaretçi yapılarıyla (özellikle parent ve children işaretçileri) uyumlu çalışmayı sağladı.

Rekürsif Hesaplamalarda Dönüş Değeri Problemi: Yaş ortalaması hesaplanırken karşılaşılan bir diğer zorluk, rekürsif fonksiyonda dönüş değerinin nasıl kullanılacağıydı. İlk yaklaşımda, her düğüm için alt ağacın yaş ortalamasını hesaplayıp döndürmeye çalıştım. Ancak bu, rekürsif çağrılarda ortalama değerlerin toplanmasına neden oluyordu ki bu matematiksel olarak hatalıydı (ortalamalar doğrudan toplanamaz). Sorunu çözmek için, fonksiyonun toplam yaşı döndürmesini ve çalışan sayısını bir işaretçi parametresi ile takip etmesini sağladım. Her düğüm ziyaret edildiğinde, düğümün yaşı toplam değere ekleniyor ve çalışan sayacı bir artırılıyordu. Rekürsif çağrılar tamamlandıktan sonra, toplam yaş değeri çalışan sayısına bölünerek gerçek ortalama hesaplanabildi.

Bu sorunların çözümü, algoritmanın doğruluğunu ve verimliliğini önemli ölçüde artırdı. Özellikle çift işaretçi kullanımı ve rekürsif hesaplamalarda değer döndürme stratejisinin değiştirilmesi, programın hem bellek kullanımında hem de hesaplama doğruluğunda iyileşme sağladı.

3-Ekran Çıktıları:

Örnek 1:

Girdi dosya içeriği:

12

A1 55 100000 NULL

B1 45 65000 A1

B2 42 60000 A1

B3 43 61000 A1

B4 44 64000 A1

C1 30 30000 B1

C2 31 31000 B1

C3 32 32000 B1

C4 28 28000 B2

C5 29 10600 B2

C6 33 33000 B3

C7 27 27000 B4

Çıktı:

```
● Calisan sayisi: 12
- Ilgili sirketin personel agaci 3 seviyeden olusmaktadir.
- Seviye 1: 1, Seviye 2: 4, Seviye 3: 7
- i=1 ise 1. seviyede en fazla calisana sahip olan kisi 4 kisi ile A1'dir.
- i=2 ise 2. seviyede en fazla calisana sahip olan kisi 3 kisi ile B1'dir.
- i=3 ise 3. seviyede en fazla calisana sahip olan kisi 0 kisi ile C1'dir.
- Tum calisanlarin yas ortalamasi: 36.58
- Sirketin odedigi aylık personel maasi: 541600.00
```

Örnek 2:

Girdi dosya içeriği:

11

A1 58 58000 NULL

B1 47 34000 A1

B2 44 33000 A1

C1 33 12000 B1

C2 34 12500 B1
C3 36 13000 B2
D1 27 9500 C3
D2 26 9400 C3
C4 28 11000 B2
C5 30 10500 B2
C6 29 10200 B2

Çıktı:

```
● Calisan sayisi: 11
- Ilgili sirketin personel agaci 4 seviyeden olusmaktadır.
- Seviye 1: 1, Seviye 2: 2, Seviye 3: 6, Seviye 4: 2
- i=1 ise 1. seviyede en fazla calisana sahip olan kisi 2 kisi ile A1'dir.
- i=2 ise 2. seviyede en fazla calisana sahip olan kisi 4 kisi ile B2'dir.
- i=3 ise 3. seviyede en fazla calisana sahip olan kisi 2 kisi ile C3'dir.
- i=4 ise 4. seviyede en fazla calisana sahip olan kisi 0 kisi ile D1'dir.
- Tum calisanlari yas ortalamasi: 35.64
- Sirketin odedigi aylık personel maasi: 213100.00
```

Örnek 3:

Girdi dosya içeriği:

20

A1 59 62000 NULL
B1 50 40000 A1
B2 51 39000 A1
C1 36 20000 B1
C2 38 21000 B1
C3 35 19000 B2
D1 28 12000 C1
D2 29 12500 C1
D3 27 11900 C2
D4 30 12200 C3
E1 24 9000 D2
E2 25 9100 D2
E3 26 9200 D4
E4 23 8800 D4
F1 22 7000 E1

F2 21 6800 E1

F3 22 6900 E3

F4 23 7050 E3

F5 24 7100 E4

F6 20 6950 E4

Çıktı:

- Çalışan sayısı: 20
 - İlgili şirketin personel ağacı 6 seviyeden oluşmaktadır.
 - Seviye 1: 1, Seviye 2: 2, Seviye 3: 3, Seviye 4: 4, Seviye 5: 4, Seviye 6: 6
 - $i=1$ ise 1. seviyede en fazla çalışana sahip olan kişi 2 kişi ile A1'dir.
 - $i=2$ ise 2. seviyede en fazla çalışana sahip olan kişi 2 kişi ile B1'dir.
 - $i=3$ ise 3. seviyede en fazla çalışana sahip olan kişi 2 kişi ile C1'dir.
 - $i=4$ ise 4. seviyede en fazla çalışana sahip olan kişi 2 kişi ile D2'dir.
 - $i=5$ ise 5. seviyede en fazla çalışana sahip olan kişi 2 kişi ile E1'dir.
 - $i=6$ ise 6. seviyede en fazla çalışana sahip olan kişi 0 kişi ile F1'dir.
 - Tüm çalışanların yaş ortalaması: 30.65
 - Şirketin ödediği aylık personel maaşı: 327500.00