



**YILDIZ TEKNİK ÜNİVERSİTESİ**

**ELEKTRİK ELEKTRONİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ**

**BLM2512 Veri Yapıları ve Algoritmalar Ödev - 3**

**Ağaçlar ile Algoritma Tasarımı**

**Öğrenci Adı:** Batuhan ODÇIKIN

**Öğrenci Numarası:** 22011093

**Dersin Öğretmeni:** Doç. Dr. Mehmet Amaç GÜVENSAN

**Tarih:** 10/05/2025

**Video Linki:** <https://youtu.be/FxHgFNJOY6c>

## 1- Problemin Çözümü

Dökümanda, bir hiyerarşi içerisinde bulunan bir çalışan düzeni verilmiş ve bu verinin yapısının korunarak saklanması ve kullanılması istenmiş. Bu problemin çözümü için general tree veri yapısı kullanılmıştır. Ağaç CEO ile başlayıp altındaki her çalışanı child olarak girilmektedir. Onların altındaki çalışanlar astı oldukları çalışanı parent olarak alarak eklenmektedir. Childları birbirine bağlamak için ise, parentta yalnızca ilk child'ın adresi tutulmaktadır, diğer childlara next sibling adresi ile ulaşılmaktadır. Aslında first childa bağlı bir linked list olarak düşünülebilir. Ağaçlarda sınırsız sayıda child ve sibling olabileceğinden ötürü node'lar arasında gezinebilmek için recursive fonksiyonlardan yararlanılmıştır.

Dökümanda istenen bilgilere ulaşılması için izlenen yöntemler altta raporlanmıştır:

Şirketin personel ağacının kaç katmandan oluştuğu: Bu problemin çözümü için ağacın root'undan başlanarak ağacın ilk çocuğuna gidilir. Her ilk çocuğa gitme hamlesinde level değişkeni 1 arttırılır. Sonrasında teker teker kardeşler gezilir, her ilk çocuğa geçme hamlesinde bu level 1 arttırılır. Bu işlem node == NULL olana kadar devam edilir. Fonksiyon bittiğinde level değişkeni return edilir, return edilen tüm değişkenler max değer ile karşılaştırılır, bu şekilde en derine gidilmiş recursive fonksiyonun değeri en derin ağaç seviyesini vermiş olur.

Her seviyede yer alan personel sayısı: yine benzer şekilde tüm node'lar gezilir ve first child işlemi olduğunda katman 1 arttırılır. Her recursive fonksiyona girildiğinde level\_array isimli bir diziye o katmanın sayısı ile aynı indexe sahip hücre 1 arttırılır. Bu sayede gezilen her node katmanı ile bu diziye kaydedilmiş olur.

Şirketin yaş ortalaması ve toplam maaş da aynı şekilde ağaç içerisinde gezilirken tüm değerlerin toplanması ile elde edilmiştir.

## 2- Karşılaşılan Sorunlar

İlk olarak bu hiyerarşiyi kurarken binary tree'deki gibi bir parent nodunda tüm childların adresini saklamaya çalıştım. Dizi gibi çeşitli yöntemler denedim fakat kullanılabilir bir yöntem elde edemedim. Sonrasında biraz araştırma sonucu linkedlist ile saklamayı deneyince istediğim sonucu elde etmiş oldum.

Yine benzer şekilde recursive fonksiyon kullanmadan ağacın dalları arasında gezmeyi denedim fakat sürdürülebilir bir sonuç elde edemeyince recursive fonksiyonlar 2 3 satırda problemi çözmüş oldum.

Katmanlardaki çalışan sayısını hesaplarken ilk başta dizi oluşturup o dizinin içerisine dolduruyordum, fakat değerlerini 0 olarak ilklendirmediğim için sebebini anlamadığım şekilde yalnızca 1 değer anlamsız bir sayı olarak gözüküyordu. İlk başka yanlışlıkla arrayın dışına çıktığımı düşündüm fakat bir hata bulamadım. Sonrasında calloc ile değerlerin içerisini 0 ile ilklendirmeyi deneyince problem çözülmüş oldu.

Son olarak ise bir katmandaki en yüksek child'e sahip kardeşi bulma probleminde ilk önce tüm kardeşlere teker teker roottan ulaşmayı, her node'un çocukların sayıp sonrasında o katmanda en yükseği bulmadı düşündüm. Fakat sonrasında bir targetLevel belirleyerek önce istediğim katmana gelene kadar recursive fonksiyonda onu eksiltip sonrasında bu katmanda tek tek kardeşleri gezerek çocuğu max olanı döndürmenin daha optimal bir çözüm olduğunu düşündüm.

### 3- Ekran Çıktıları

```
PS C:\Users\batuh\OneDrive\Masaüstü\YTU_CE\2_2\Veri Yapıları ve Algoritmalar\Ödevler\Ödev 3> c
1 ve Algoritmalar\Ödevler\Ödev 3\" ; if ($?) { gcc 22011093.c -o 22011093 } ; if ($?) { .\2201
A1
  B1
    C1
    C2
    C3
  B2
    C4
    C5
  B3
    C6
  B4
    C7
Height of tree: 3
level(1):1
level(2):4
level(3):7
Average of age 36
Average of age 100000
Sibling with most children at level 1: B1 has 3 children.
Sibling with most children at level 2: B1
PS C:\Users\batuh\OneDrive\Masaüstü\YTU_CE\2_2\Veri Yapıları ve Algoritmalar\Ödevler\Ödev 3>
```