

## Ödev 5: Ağaç Yapıları - İkili Arama Ağacı Yapısı (03.12.2019)

Sektörde veya bilgisayar/algorithm problemlerinde kullanım alanlarını araştırınız. Örnek bir bilişim senaryosunu gerçekleyen programı yazınız.

*Hint:* İkili arama ağacı için kullanılabilecek örnek yapı:

```
typedef struct binary_tree_nodes{
    int value;
    struct binary_tree_nodes *left, *right;
}Node;
//...
Node *root=NULL; //Initial state: the first node of the tree
```

- Veri yapısı fonksiyonları ayrı kaynak kod dosyasında tanımlanacaktır. Gerekli olduğunda senaryo fonksiyonları içerisinde çağırılacaktır.

```
#include "myBST.h"
```

- Veri yapısının gerçekleştirilmesi gereken fonksiyonlar:
  - **Ekle:** İlgili bilgi ağaçta uygun yere eklenecektir. Aynı veri birden fazla eklenebilir olmalıdır.
  - **Sil:** İlgili bilgi (birden fazla olduğu takdirde tümü / ilki / sonuncusu) silinecektir.
  - **Listele:** Ağaçtaki elemanlar farklı şekillerde dolaşılabilir:
    - **Önce Gezinti (Preorder) Dolaşma**
    - **Ara Gezinti (Inorder) Dolaşma**
    - **Sonra Gezinti (Postorder) Dolaşma**
  - **Say:** Veri yapısının içerdiği toplam eleman sayısı ağaç taranarak hesaplanacaktır (tüm/eşsiz).  
Örnek: 3,2,1,2,3,4,3,5,6,7 sayıları veri yapısına yerleştirilmiş olsun.  
X adet yaprak ve 7 adet eşsiz değer olmak üzere 10 eleman vardır.
  - **Dengele:** Ağacın sahip olduğu eleman sayısına göre minimum derinliği elde etmek için ağaç yeniden oluşturulacaktır.  
Örnek: 3,1,2,4,5,6,7 sayılarının sırasıyla *ikili arama ağacı* yapısına eklendiğini düşünün; derinlik nedir? Bu sayılar *mükemmel ikili arama ağacı*na yerleştirilmiş olsaydı derinlik ne olurdu?
  - **Göster (Bonus):** Ağaç yapısı (ebeveynler ve çocuklar anlaşılacak şekilde) ekranda sunulacaktır.
- Senaryo, kullanıcı etkileşimi ile menü üzerinden gerçekleştirilebilir. Problemin tanımına göre bir simülasyon da hazırlanabilir. Yalnızca test fonksiyonları üzerinden demo yapılması yetersizdir.

---

Uygulamalar haftaya göre değişkenlik gösteren yüzdeler ile *demo*, *performans* (kod düzeni, kodlama standartları, açıklama satırları, ...) ve *quiz* olmak üzere üç kriter üzerinden değerlendirilir. Kaynak kod dosyası veya proje dosyaları ve derlenmiş .exe dosyası (<OgrNo>\_odev<#><Sube>.xxx), örnek koşu/çalıştırma sonuçlarını içeren bir adet çıktı dosyası (<OgrNo>\_odev<#><Sube>output.txt/jpg: konsol metni veya görsel arayüz programları için ekran görseli) olmak üzere dosyalar, <OgrNo>\_VeriYapLab<Sube>1920GUZ\_Odev<#>.zip isimlendirme formatında yükleme alanına yüklenir. Yükleme hatalarına ceza puanı uygulanır.

Örnek İsimlendirme: 152120081026\_VeriYapLabC1920GUZ\_Odev9.zip

**Sisteme yüklenmeyen veya yüklenip laboratuvar sırasında gösterilmeyen çalışmalar geçersiz sayılır.**

---