

## Ödev 4: İkili Ağaç Yapısı (06-07.11.2017)

İkili arama ağacı oluşturan konsol veya GUI programını yazınız.

➤ Ağaç *pointer* tabanlı olup veri tam sayılardan oluşacaktır.

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

```
typedef struct binary_tree_nodes{
    int value;
    struct binary_tree_nodes *left, *right;
}Node;
```

```
//Initial state: first node of the tree
```

```
Node *root=NULL;
```

- Menüde alt işlemler olacaktır ve fonksiyonel yapı kullanılmalıdır; bu işlemlerin herhangi biri yapılmak istendiğinde ilgili fonksiyon çağırılmalıdır.
  - **Ekle:** Kullanıcıdan istenen sayı ağaçta uygun yere eklenecektir.
  - **Sil:** İlgili sayı (birden fazla olduğu taktirde tümü) ağaçtan silinecektir.
  - **Listele:** Ağaçtaki elemanların hangi şekilde listeleneceği kullanıcıya sorulmalıdır.
    - **Önce Gezinti (Preorder) Dolaşma**
    - **Ara Gezinti (Inorder) Dolaşma**
    - **Sonra Gezinti (Postorder) Dolaşma**
  - **Say:** Toplam eleman sayısı ağaç taranarak hesaplanacaktır.
  - **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 (*nodeların ebeveynleri ve çocukları anlaşılacak şekilde*) ekranda sunulacaktır.
- Tüm program olabildiğince kullanıcı dostu olmalıdır.
- Her fonksiyonun zaman karmaşıklığı ve hafıza kullanımı hakkında bilgi sahibi olunmalıdır.

Uygulamalar dys'ye yüklenir, aynı zamanda laboratuvarında gösterilir. Sisteme yükleme yapmayan veya yükleme yapıp ilgili tarihteki laboratuvara katılmayan öğrencilerin ödevleri geçersiz sayılır. İlgili projeyi, sıkıştırılmış dosyanın içine projenin derlenmiş halini de ekleyerek <OGRNO>\_VeriYapLab1718Odev4.rar ismi ile dys'ye yükleyiniz.