Ödev 6: Ağaç Yapıları - *Splay* Ağacı Yapısı (10.12.2019)

Splay ağaç yapısı ve fonksiyonlarını *implement* ediniz.

- *Splay* ağaçları ikili arama ağacının özelleşmiş bir türü olarak tanımlanmaktadır. Bu doğrultuda Ödev5 kapsamında gerçekleştirilen ikili arama ağacı yapısından yararlanılabilir.
  ```
  #include "myBST.h"
  ```

- Herhangi bir senaryoya veya teknik örneğe ihtiyaç yoktur. Ancak yapıldığında BONUS puan kazanılabilir. Çalışmanın yalnızca veri yapısının birim testi (*Bkz. Unit Test*) niteliğinde olması yeterlidir.

- Veri yapısı fonksiyonları ayrı kaynak kod dosyasında tanımlanacaktır. Başka bir kaynak dosyadaki test fonksiyonları ile kullanıcı girişi üzerinden test edilecektir.
  ```
  "mySplayTree.h"
  ```

- Veri yapısının gerçekleştirmesi gereken fonksiyonlar:
  - Tüm ikili arama ağacı yapısı fonksiyonları kullanılabilir olmalıdır.
  - Hafızaya erişim veya etkileşim ile oluşan **zig** / **zag** durumları kodlanmalıdır.
  - Ödev6 kapsamında yalnızca okuma/arama durum senaryoları için **zig** / **zag** operasyonlarının gerçekleştirilmesi yeterlidir.

- Erişim öncesi, iterasyonlar arası ve erişim sonrası ağacın yapısında oluşan değişiklikler gösterilmelidir.
  Örnek: ⌘ *zig-right* ⌘ *zag-left* ⌘
  ⌘ Her bir gösterim aşaması

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<#><Şube>.xxx**), örnek koşu/çalıştırma sonuçlarını içeren bir adet çıktı dosyası (**<OgrNo>_odev<#><Şube>output.txt/jpg**: konsol metni veya görsel arayüz programları için ekran görseli) olmak üzere dosyalar, **<OgrNo>_VeriYapLab<Şube>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.