

VERİ YAPILARI DERSİ 2.ÖDEV RAPORU

Ödev Grubu(B)

Öğrenci Adı: Ahmed

Öğrenci Soyadı: Hachüseyin

Öğrenci No: G211210586

Öğrenci Adı: Kays

Öğrenci Soyadı: Alsaran

Öğrenci No: G211210584

1. **"Node"** sınıfı, ikili arama ağacının düğümlerini temsil eder. Her düğüm bir veri içerir ve sol ve sağ alt ağaçlara işaret eden pointerlara sahiptir.
"BinaryTree" sınıfı, ikili arama ağacını oluşturan, üzerinde işlemler gerçekleştiren ve verileri çeşitli şekillerde dolaşarak işleyen fonksiyonları içerir.
2. Bu kod, ikili arama ağaçlarını **"DoublyLinkedListBinaryTree"** adında bir çift yönlü bağlı liste üzerinde depolayan bir veri yapısı tanımlar. Bu veri yapısı, ikili arama ağaçlarını listenin sonuna ekleyebilir ve listeden çıkarabilir. Ayrıca, listenin boyutunu, belirli bir indexteki ikili ağacı ve listenin başlangıcındaki düğümü almak gibi işlevleri de gerçekleştirebilir.
3. **"NodeList"** Yapısı: Bu yapının int türünde bir veri, bir önceki düğümü (prev) ve bir sonraki düğümü (next) tutan üyeleri bulunur. Bu yapı, bağlı listeye ait bir düğümü temsil eder.
"DoublyLinkedListForIntegral" Sınıfı: Bu sınıf, çift yönlü bağlı listenin işlevlerini tanımlar. Tamsayı değerlerini listenin sonuna eklemek, çıkarmak, boş olup olmadığını kontrol etmek, boyutunu döndürmek, belirli bir indexteki değeri almak gibi işlemleri içerir.
4. **"NodeListForStack"** Yapısı: Bu yapının Stack türünde bir veri, bir önceki düğümü (prev) ve bir sonraki düğümü (next) tutan üyeleri bulunur. Bu yapı, bağlı listeye ait bir düğümü temsil eder.
"DoublyLinkedListForStack" Sınıfı: Bu sınıf, çift yönlü bağlı listeyi yöneten işlevleri içerir. Stack veri yapısını listenin sonuna eklemek, çıkarmak, boş olup olmadığını kontrol etmek, boyutunu döndürmek, belirli bir indexteki değeri almak gibi işlemleri gerçekleştirir.

5. "**Stack**" veri yapısı tanımlar ve bu yığıt üzerinde temel yığıt işlemlerini gerçekleştiren işlevleri içerir.

"**NodeStack**" sınıfı, yığıtın düğümlerini temsil eder. Her düğüm, bir veri içerir ve bir sonraki düğüme işaret eden bir next pointerına sahiptir

"**Stack**" sınıfı, yığıtın işlevlerini tanımlar.

6. "**Metin**" dosyasından satır okuyarak işlem yapar. Her satır, sayıları içerir ve bu sayıları işleyerek ikili arama ağaçları oluşturur. Ardından bu ağaçlar arasından en büyük yüksekliğe sahip olanını bulur ve bu ağacın düğüm değerlerini postorder şeklinde dolaşarak değerlerini ekrana yazdırır.

"**processLine(string line)** ": Verilen bir satırı işler, sayıları çıkarır ve bunları bir yığıt yapısı kullanarak ayrı yığıtlara gruplar. Yığıtlar, belirli koşullara uygun olarak ayrılır.

"**createBinaryTree(Stack stack)** ": Verilen bir yığıt yapısından bir ikili arama ağacı oluşturur. Yığıttaki sayılar sırasıyla çıkarılarak ikili arama ağacına eklenir.

"**findMaxHeightTree(DoublyLinkedListBinaryTree& trees)** ": Verilen çift yönlü bağlı listedeki ikili arama ağaçları arasında en yüksek boyuta sahip olanını bulur.