

Ödevde ikili arama ağacı kullanımı, düğümler ve pointerların daha iyi anlaşılması hedeflenmiştir. Bu doğrultuda yaptığım projede Node ve BinarySearchTree olmak üzere iki sınıf kullandım.

Node sınıfındaki değerleri public olarak tutmayı tercih ettim bunun sebebi Node sınıfının herhangi bir kompleks yapıda olmaması ve değerlerin dışarıdan değiştirilmesinin sınıf içinde bir soruna yol açmayacak olmasıdır. Bu sınıf; left ve right olmak üzere Node sınıfından iki pointer, bir tane veriyi tutacak T sınıfından (T, template'dir) data değişkeni ve integer türünde childCount değişkenine sahiptir. Node sınıfının constructor'ında data değişkeninin değeri dışarıdan alınır, geriye kalan değerler ise başlangıç değeri için NULL ve 0 atanır. Node sınıfının bir de destructor'ı bulunur. Bu methotta Node nesnesinin heap alanından silindiğine dair bilgi, console ekranına yazdırılır.

BinarySearchTree sınıfı root adında Node nesnesinin pointer'ını tutar. Bu root adresi sayesinde ağacın diğer dallarına ulaşabiliriz. Bu sınıfta insert adlı method bulunur. Bu method girilen değerleri sırasıyla ağaca uygun yere yerleştirilir. Bu method recursive işlem yapar. Sıraya göre önce ağacın kökünden başlayarak ağaca eklenmek istenen değer, düğümden büyükse düğümün sağ koluna eğer küçükse sol koluna olacak şekilde yerleştirilir. Bu yerleşim şekli aranan değer bulunmasında bize hız kazandırır. Her ekleme işleminden sonra düğümlerde bulunan childValue değeri güncellenir. Bu değer düğümlerin ne kadar yaprağa sahip olduğunu gösterir. Ağaçtaki değerler postorder sırasıyla okunur ve ekrana printAll() fonksiyonu ile yazdırılır. Düğümlerdeki childCount değeri de postorder şekilde dolaşarak güncellenir. Postorder sırasına göre ağacın kökünden başlanarak yaprağına doğru gidilir. Gidilebildiği kadar sola gidilir. Eğer solda değer yoksa sağa gidilir. Bu şekilde ağaçtaki tüm düğümler okunmuş olur. Silme işlemi olarak sol yapraktaki en büyük değer kök yapılması tercih edilmiştir. BinarySearchTree sınıfında bir de findMax isimli method bulunur. Bu method ağaçtaki en büyük değeri bulmaya yarar. Bu method da sınıftaki diğer methodlar gibi recursive yapıdan faydalanır. BinarySearchTree sınıfından oluşturulan bir nesne heap alanından silinirken Node nesneleri de silinir. Bu silme işlemi makeEmpty methodu ile gerçekleştirilir. Bu method da recursive işlem yapar.

Program iki score ve bir round değişkeni tutar. Burada iki BinarySearchTree nesnesi heap alanında oluşturulur ve text dosyalarındaki sayılar bu ağaçlara eklenir. Node nesnelerindeki child sayıları toplanır ve en küçük değere sahip olan nesne ilk turu kazanır, score değişkenine bir eklenir. Beş tur kazanan ya da round sayısı yirmi olduğunda kim öndeyseniz oyunu o ağaç kazanır.

Genel olarak bu ödevden öğrendiklerim; C++ Syntax'ı, C++ Sınıf yapısı, Pointers, Bellek Yönetimi ve bağlı liste kavramları oldu.