HW06 Report-Ahmet Emin Kaplan-131044042

PART1

Bizden istenen problemin çözümünün recursive olması gerektiğini düşündüm bir ağacın bütün dallarina iterative olarak erişmek recursive olarak erişmekten daha zor olduğu için problemi recursive dizayn ettim.Text formatinda girilen herhangi bir stringin her bir karakteri sirasiyla recursive yazdiğim metoda gönderilir.charToHuffmanCode() metodu recursive olarak karakteri huffmantree de arar .Metodun 2 base case i var bunlardan biri aranan karakterin bulunmasi diğeri ise bulunamaması.Huffman Treede karakterler leaf olan nodelarda olduğu için eğer bir node leaf ise o nodun içersindeki karaktere bakılır .Karakterin mevcut olup olmamasina göre durum handle edilir ve Karakterler huffman koduna çevrilir.

PART2

BinarySearchTree üzerinde iterator ascending order biçimde gezinebilmesi için iteratörün başlangıç konumunu treenin en sol ve en alttaki(en küçük) elaman olarak initiliaze ettim.Tabiki initiliaze ederken her bir parent node tanımlamış olduğum stackte saklanmakta.Her seferinde en küçük elemanin return edibilmesi için stackten çektiğim parent node un sağ child inin minimum elemanlarina erişilirken stack güncellenir.Eğer stackte hiç eleman yoksa ve nextItem isimli member null göstermekte ise agacin sonuna gelinmiştir bu hususta hasNext() metodu false döndürecektir.Iteratorun diğer bir sahip olduğu remove() metodu için binarySearchTree nin sahip olduğu metodu lastReturnedItem isimli memberla cagirdim.

PART3

Son dönemde üst üste gelen ödevlerin yoğunluğu sebebiyle bu parti yetiştiremedim.

Bu ödev sayesinde huffman tree hakkında daha fazla bilgi edinmiş oldum.BinarySearchTree üzerinde çalışan bir iterator yazmanin bana object oriented metodolojiyi daha iyi kavrattığını düşünüyorum.