Ödeve başlamadan önde draw.io sitesinde aşşağıda gördüğünüz şemayı çizdim, kuracağım yapıyı bu şekilde yapmayı planladım.

A diagram of a computer program

Description automatically generated with medium confidence

1. AVL Agac yapısı oluşturma :Genel yapımı kurduktan sonra ilk iş olarak AVL yapısı için AVLAgacı ve AVLDugum isimli iki class oluşturarak, ihtiyacım olan fonksiyonları oluşturdum. Bu işlemleri bitirdikten sonra artık bir ağaç oluşturabiliyor ve ona eleman ekleyebiliyordum.
2. Yıgın yapısı oluşturma: Şimdi bizden istenen 2. Bir şey ise oluşturduğumuz avl ağacının yapraklarındaki int değerlerinin bir yığın da olmasıydı bu yüzden yığın yapımı oluşturdum.

Artık ağaç ve yığın yapılarını oluşturduğuma göre her iki class tan değişken tutacak list’in dugumunu ve bu dugu içerecek list yapısını oluşturmaya geçebilirdim, bu işlemide yaptım. Burada yaptığım bazı şeylere dikkat çekmek istiyorum

En küçük ve en büyük yığınları çıkartma işlemini adım adım şu şekilde yaptım önce enKucuk fonksiyonu nu oluşturdum ve bu fonksiyon min değerini bulucak şekilde yaptım doğru çalıştığını test ettikten sonra yığından eleman çıkarma eğer yığında eleman kalmamış isede o liste dugumunu silmeyi ayarladım, silme işlemi için pointer ile silinecek liste dugumunun bir önceki dugumu tuttum bu sayede silme işlemini kolaylıkla gerçekleştirdim, tek dikkat etmem gereken spesifik durum 1. liste silinecek eleman olma durumu idi bunun içinde en küçük bulunduğu zaman bunun 1. Yığının sayısı ile bulunan min sayısının karşılaştırıp eşit ise direk 1. Dugumu sildim değil ise normal algoritma çalıştı. enKucuk fonksiyonu bittikten sonra kopyala yağıştır yapıp min yerine max yazmak ve birkaç if de değişiklik ile enBuyuk fonksiyonunu da oluşturdum sonra bu iki fonksiyonu bir while içinde çalıştırıcak ve bir harf çıkarılana kadar yani bir yığın tamamen boşalana kadar while içerisinde dönecek olan harfCıkar fonksiyonunu oluşturdum. enKucuk veya enBuyuk fonksiyonu eper bir dugum siler ise 1 değeri return edicke (döndürecek) ve while göngüsü burada bitecek. Devamında ise harfleri yazdıran fonksiyon çalışacak sonra harfCikar fonksiyonu yeniden çalışacak bu işlem ise 1 harf kalana kadar devam edecek.

Projede yığından sayılar çıkarıyoruz (en küçük, en büyük) ve ilk biten yığının ait olduğu list yapısı siliniyor, terminalde ise bu yapının sahip olduğu char karakterinin kaybolması ile biz bu durumu gözlemleyebiliyoruz lakin bu işlem sırasında birçok yığın yapısı da bozulmuş oluyor bizim bu yığın yapılarını yeniden oluşturmamız gerekiyor. İlk akla gelen her bir harf çıkarıldığında tüm yığın yapılarını yeniden oluşturmak ama bu oldukça saçma olurdu şöyle düşünün benim 500 adet ağaç yapım dolayısı ile 500 adet yığın’ım var lakin ve 15 adet yığın da çıkarma yaptım, bu durumda yalnızca değişiklik yaptığım bu 15 yığını güncelleyip diyer 485 yığın’a hiç dokunmamak, bana büyük bir performans kazandıracaktır. İşte bunun için yıgınGuncelmi diye bir değişken oluşturdum ilk başta bu değişkeni bool tipinde yapıp sadece çıkan verileri geri ekleyeyim diye de ekleme fonksiyonu nu sağdan çalışacak şekilde dizayn etmiştim ama o anda düşünemediğim şey bu eklemenin yığın sırasını bozacağı olmuştu. Sonrada bunu düzeltmek için oluşturduğum yıgınGuncelmi değişkenini int yaptım, her çıkarma yapıldığında bir arttırdım ve sadece 0’a eşit olmayan yani yığını değiştirilmiş yapıların yığınını güncelleyecek şekilde algoritmamı kurdum bu nokta da, yığın güncelle algoritması ilk başta yığın oluşturma algoritması ile benzer çalışıcak tek farkı ise her yaprağa olaştığında yıgınGuncelmi değişkenini arttıracak ne zaman ki bu değişken yığın uzunluğuna eşit olur işte o zaman gelen değerleri yığına ekleyecek şekilde algoritmamı kurdum bu sayede yalnızda değişen yığınları güncelleyecek yapım oluşmuş oldu. Burada yığına ekleme yaparken ekle2 fonksiyonu yaptım çünkü uzunluk değişkenini ekle fonksiyonu içerisinde değiştiriyor idim. Bu yaptıklarım ile de yığın yapısını ve güncellenmesini halletmiş oldum.