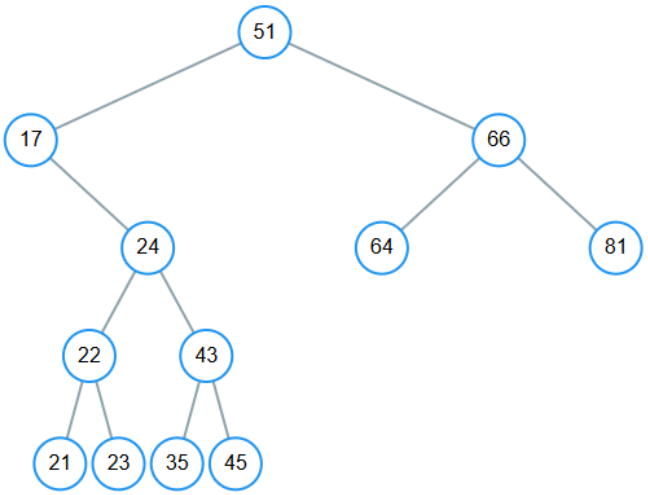
GİRESUN ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
VERİ YAPILARI DERSİ FİNAL SINAVI

Adı – Soyadı – Numarası:

Soru 1: Aşağıda verilen ikili ağaç için (a) kök önde (preorder), (b) kök ortada (inorder) ve (c) kök sonda (postorder) gezildiğinde oluşan çıktıyı yazınız. Kök düğüm 51 numaralı düğümdür. (d) Kök ortada gezmenin çıktısı bize neyi verir?



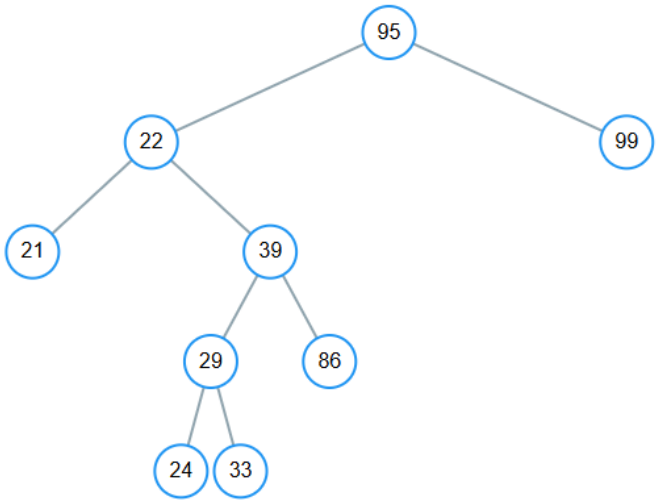
Preorder: 51 -> 17 -> 24 -> 22 -> 21 -> 23 -> 43 -> 35 -> 45 -> 66 -> 64 -> 81

Inorder: 17 -> 21 -> 22 -> 23 -> 24 -> 35 -> 43 -> 45 -> 51 -> 64 -> 66 -> 81

Postorder: 21 -> 23 -> 22 -> 35 -> 45 -> 43 -> 24 -> 17 -> 64 -> 81 -> 66 -> 51

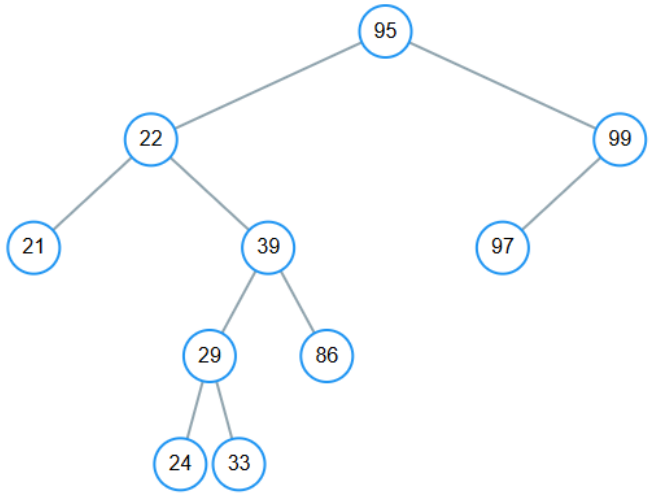
Eğer dolaşılan ağaç bir ikili arama ağacı ise, in-order traversal, düğüm değerlerini küçükten büyüğe sıralı olarak verir.

Soru 2: Aşağıda verilen ikili arama ağacına, belirtilen işlemler sırayla uygulandıktan sonraki halini çiziniz. (a) ekle 97, ekle 45, sil 39, ekle 39. (b) Silme işlemi sırasında ağacın yapısı hangi durumlarda bozulabilir? Ne tür önlem alınabilir?

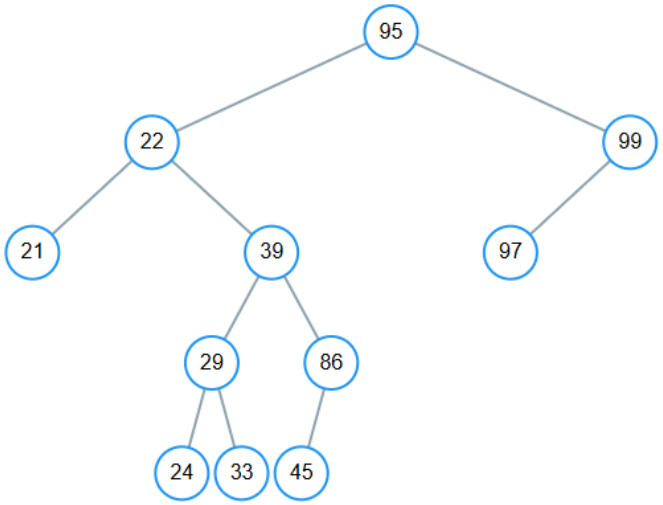


Başarılar.. Dr. Öğr. Üyesi Sercan KÜLCÜ 09.01.2025

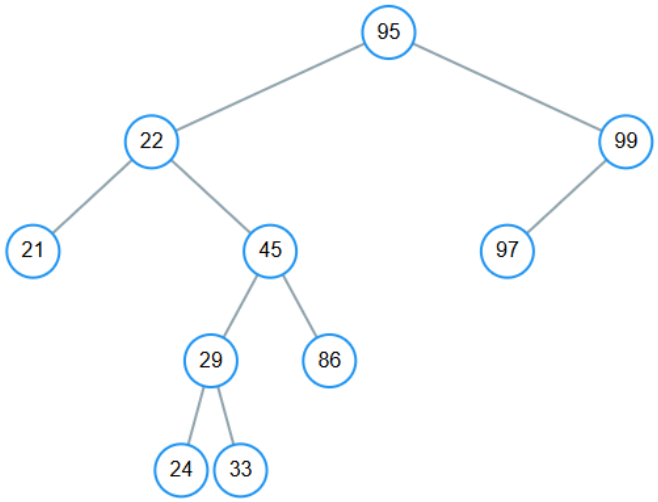
97 eklendikten sonra



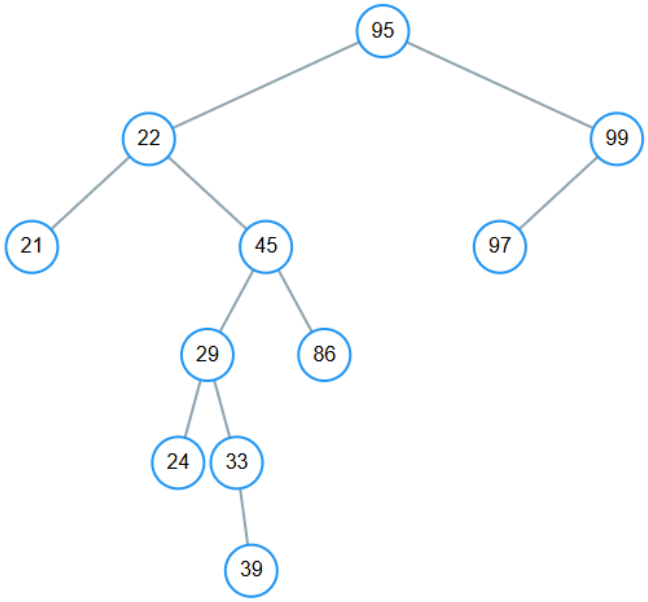
45 eklendikten sonra



39 silindikten sonra

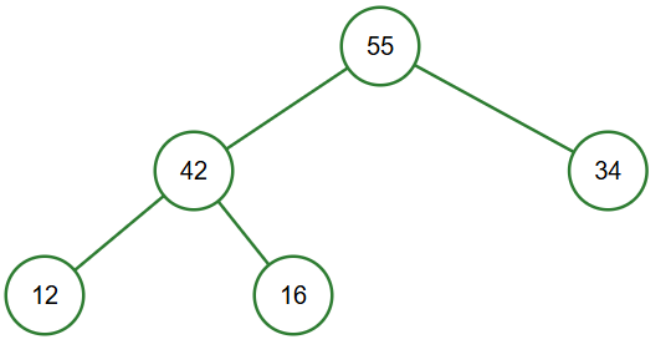


39 eklendikten sonra

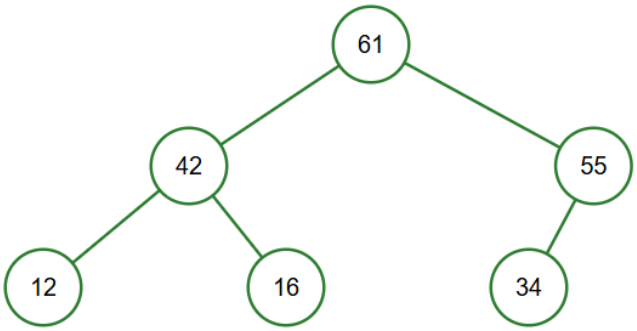


İki çocuklu düğümün silinmesi durumunda, ağacın yapısı bozulabilir. Çünkü silinen düğümün yerini alacak uygun bir düğüm bulunmalıdır. İkili arama ağacı özelliğini korumak için, genelde sağ alt ağacın en küçük elemanı (in-order successor) veya sol alt ağacın en büyük elemanı (in-order predecessor) silinen düğümün yerine geçirilir.

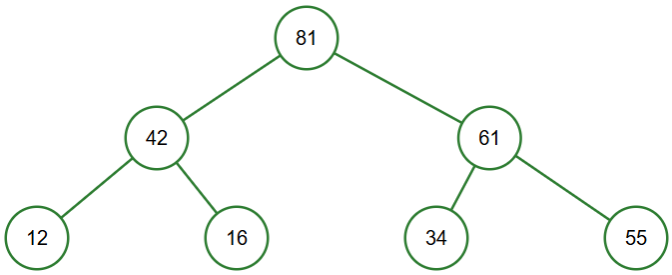
Soru 3: Aşağıda verilen maksimum ikili yığına (max-heap) verilen işlemler uygulandıktan sonraki halini çiziniz. (a) ekle 61, ekle 81, sil (extract max), sil (extract max), ekle 61. (b) Maksimum yığını hangi işlem veya işlemlerde kullanmak avantajlıdır.



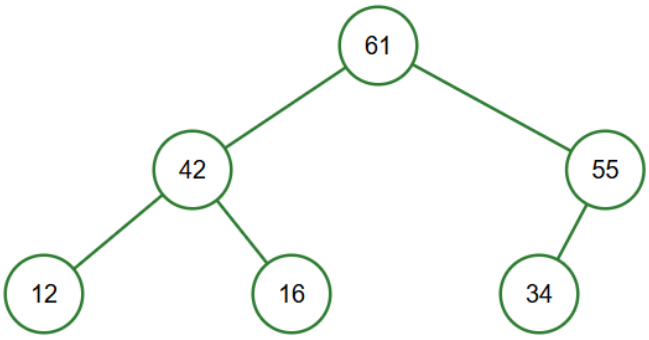
61 eklendi



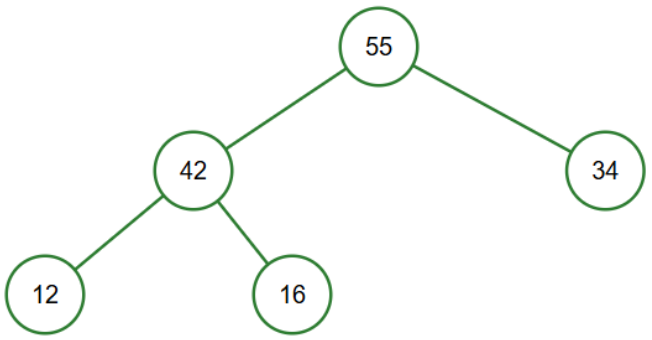
81 eklendi



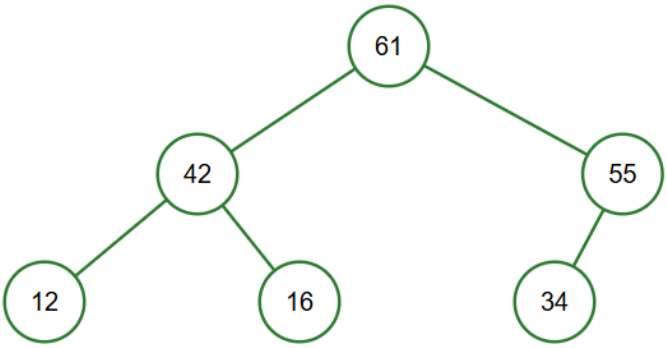
Sil (extract max)



Sil (extract max)



61 ekle



Kök düğüm (root), her zaman en büyük eleman olduğundan, maksimum eleman O(1) zaman karmaşıklığıyla alınır. Büyük veri kümelerinde en büyük değeri hızlıca almak için kullanılabilir. Maksimum öncelikli elemanları hızlıca işlemek için kullanılır.

Soru 4: Verilen elemanları (19, 29, 39, 49, 59, 69, 79, 89) boyutu 10 olan bir hash tablosunda saklamak için, çakışmayı (collision) minimumda tutacak (a) hash fonksiyonu öneriniz. (b) Çakışma oranını hesaplayınız.

h(x) = (x / 10) mod 10

Bu fonksiyon her elemanı hash tablosunun farklı bir hücresine eşler, çakışma sayısı sıfırdır.

Tablo indeksleri: 1, 2, 3, 4, 5, 6, 7, 8

h(x) = x mod 7

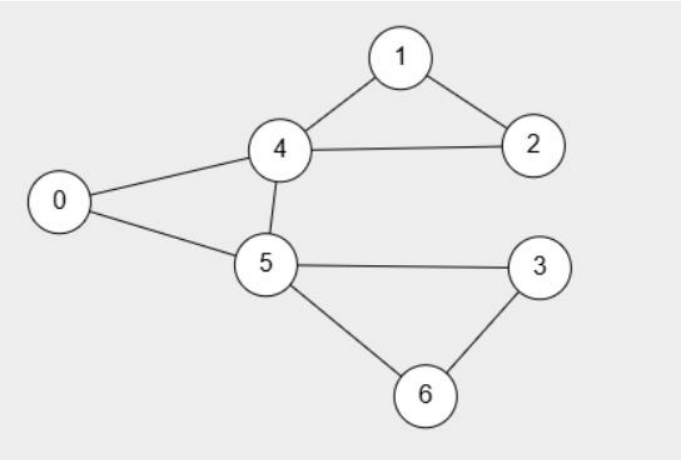
Eleman (x)	**19**	**29**	**39**	**49**	**59**	**69**	**79**	**89**
X mod 7	**5**	**1**	**4**	**0**	**3**	**6**	**2**	**5**

Çakışma yalnızca bir yerde (19 ve 89) oluşur.

Tablo indeksleri: 5, 1, 4, 0, 3, 6, 2, 5

mod 7 kullanıldığı için tablonun tamamını eşleyemez.!

Soru 5: Aşağıda verilen yönsüz çizgeyi (a) komşuluk matrisi ile gösteriniz. (b) Derinlik Öncelikli (DFS) ve (c) Genişlik Öncelikli (BFS) arama ile gezildiğinde oluşan yolu (path) yazınız. Komşuların seçilme sırası küçükten büyüğe doğru olacaktır. Aramaya 0 nolu düğümden başlanacaktır. (d) DFS ve BFS hangi durumda aynı çıktıyı üretir?



GİRESUN ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
VERİ YAPILARI DERSİ FİNAL SINAVI

Adj. Matrix

0	0	0	0	1	1	0
0	0	1	0	1	0	0
0	1	0	0	1	0	0
0	0	0	0	0	1	1
1	1	1	0	0	1	0
1	0	0	1	1	0	1
0	0	0	1	0	1	0

Adj. List

0 -> 5, 4
1 -> 4, 2
2 -> 4, 1
3 -> 6, 5
4 -> 5, 0, 1, 2
5 -> 0, 4, 3, 6
6 -> 3, 5

BFS: 0 → 4 → 5 → 1 → 2 → 3 → 6

DFS: 0 → 4 → 1 → 2 → 5 → 3 → 6

Eğer bir ağaç veya çizgede her düğümün yalnızca tek bir çocuğu varsa, DFS ve BFS aynı sırada düğümleri ziyaret eder. Eğer tüm düğümler aynı düzeydeyse ve birbirine dallanmadan bağlıysa, DFS ve BFS aynı sırada düğümleri ziyaret eder.

Başarılar.. Dr. Öğr. Üyesi Sercan KÜLCÜ 09.01.2025