

CENG 201 Veri Yapıları 7: Heap, 2-3 Ağaçları ve B-Ağaçları

Öğr.Gör. Şevket Umut ÇAKIR

Pamukkale Üniversitesi

Hafta 7

Anahat

① Yığın/Heap

Ekleme

Silme

② 2-3 Ağaçları

Ekleme

Silme

③ B-Ağaçları

Ekleme

Silme

Öncelik Kuyruğu(Priority Queue) Tanım

Tanım

Öncelik kuyruğu ekleme(insert) ve en küçüğü silme(deleteMin) işlemlerine sahip bir veri yapısıdır. deleteMin işlemi normal kuyruk yapısındaki dequeue işlemi gibi çalışır ama her zaman en küçük değere sahip elemanı listeden çıkarır.

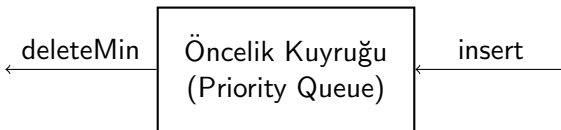


Figure: Öncelik kuyruğu modeli

İkili Yığın/Binary Heap

Tanım

Öncelik kuyruklarının gerçekleştirilmesi için ikili ağaç yapısında verimli bir veri yapısıdır.

Heap aşağıdaki özellikleri sağlar:

- Heap tam ikili ağaç(complete binary tree) yapındadır
- Bir düğümdeki anahtar değeri çocuklarının anahtar değerinden daha küçüktür(Minimum Heap)
- Tam ikili ağaç yapısında olduğu için dizilerle temsil edilebilir, sol çocuk: $2i$, sağ çocuk: $2i + 1$
- Ekleme ve silme işlemlerinde bozulan heap yapısını oluşturmak için **Heapify** işlemi yapılır

Heap Örneği

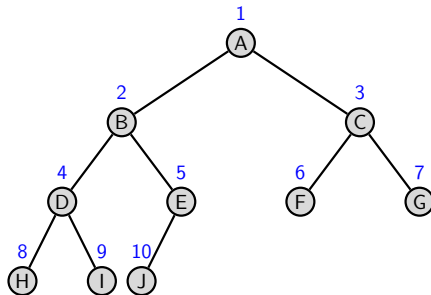


Figure: Heap Örneği

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Değer		A	B	C	D	E	F	G	H	I	J			

Ekleme

- Ekleme işlemi tam ikili ağacın en alt seviyesinin en sağına gerçekleştirilir
- Eğer Heap özelliği bozulmuşsa ebeveyn ile eklenen çocuk yer değiştirir, bu işlem köke kadar devam eder

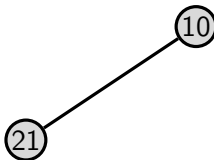
Ekleme Örneği: 10

10, 21, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim

10

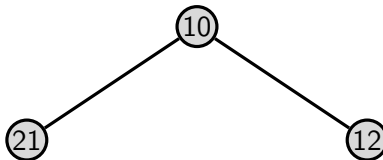
Ekleme Örneği: 21

10, 21, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



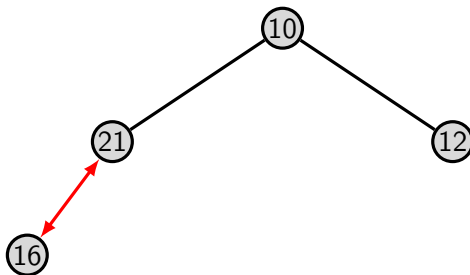
Ekleme Örneği: 21

10, 21, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



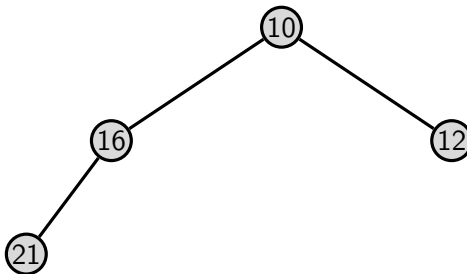
Ekleme Örneği: 16

10, 21, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



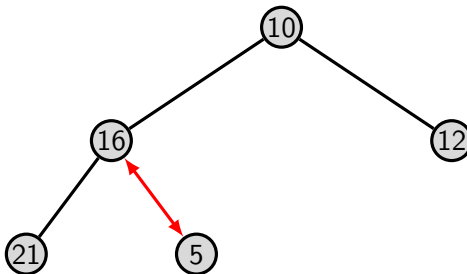
Ekleme Örneği: 16

10, 21, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



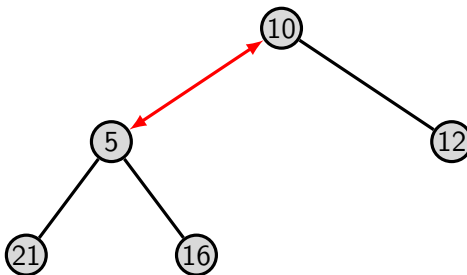
Ekleme Örneği: 5

10, 21, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



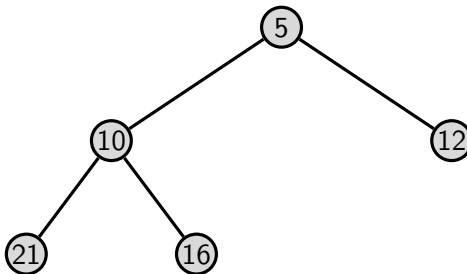
Ekleme Örneği: 5

10, 21, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



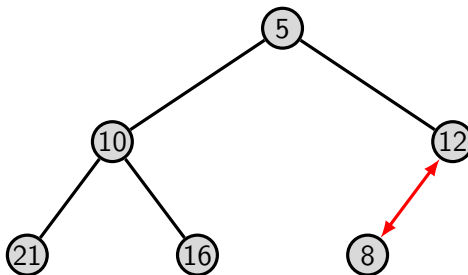
Ekleme Örneği: 5

10, 21, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



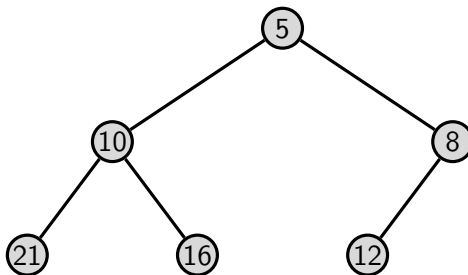
Ekleme Örneği: 8

10, 21, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



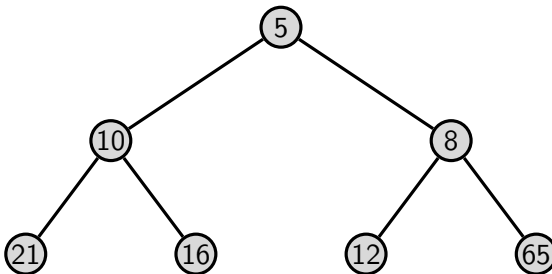
Ekleme Örneği: 8

10, 21, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



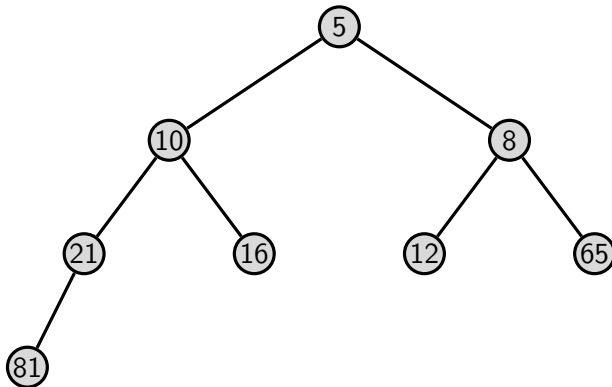
Ekleme Örneği: 65

10, 21, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



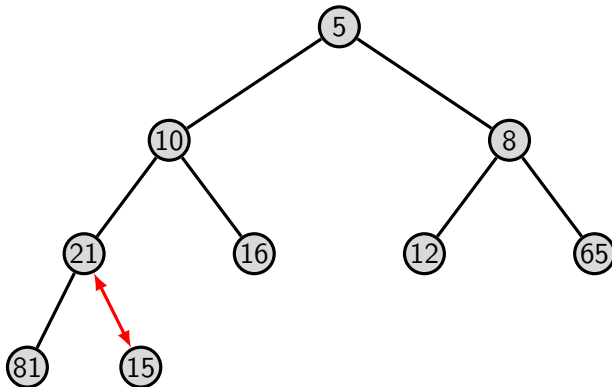
Ekleme Örneği: 81

10, 21, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



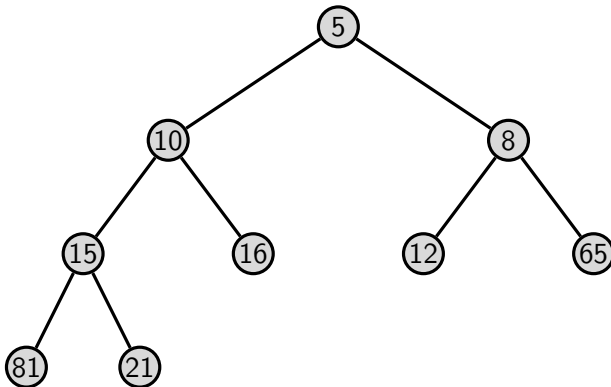
Ekleme Örneği: 15

10, 21, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



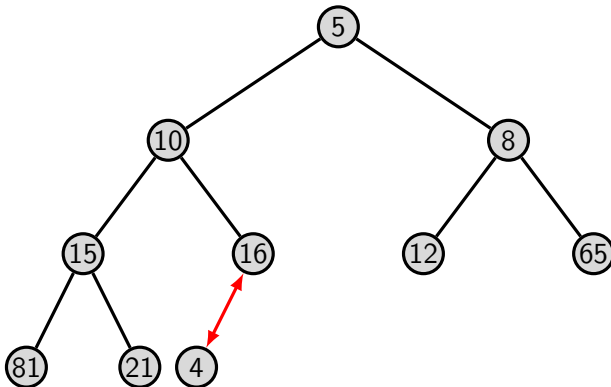
Ekleme Örneği: 15

10, 21, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



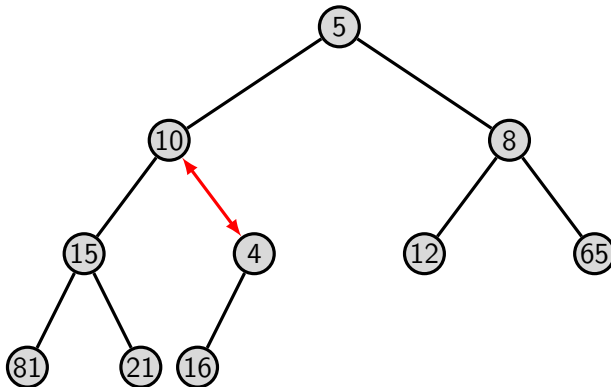
Ekleme Örneği: 4

10, 21, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



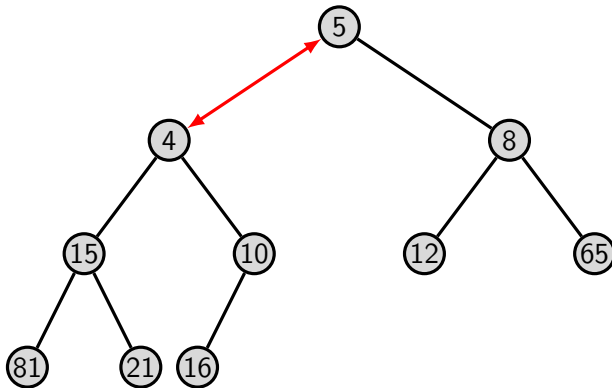
Ekleme Örneği: 4

10, 21, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



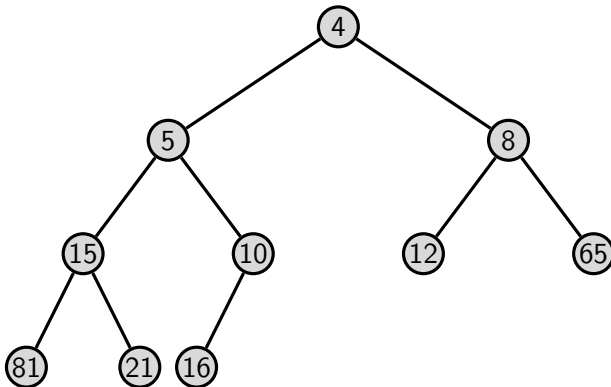
Ekleme Örneği: 4

10, 21, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



Ekleme Örneği: 4

10, 21, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim

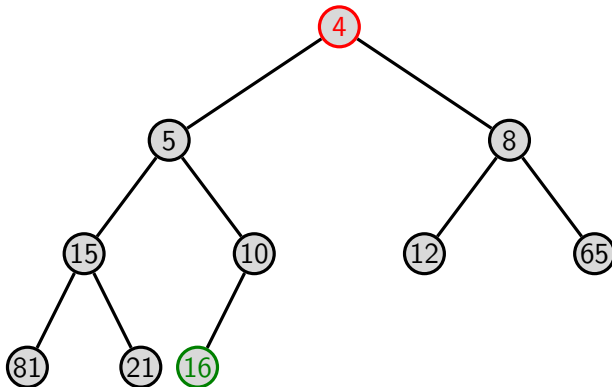


Silme

- Silme işlemi kökten yapılır(deleteMin)
- Kökteki elemanın yerine en alt seviyenin en sağındaki eleman çıkartılıp yazılır
- Heap özelliğini sağlamak için aşağı doğru çocuklarından hangisi küçükse onunla yer değiştirir

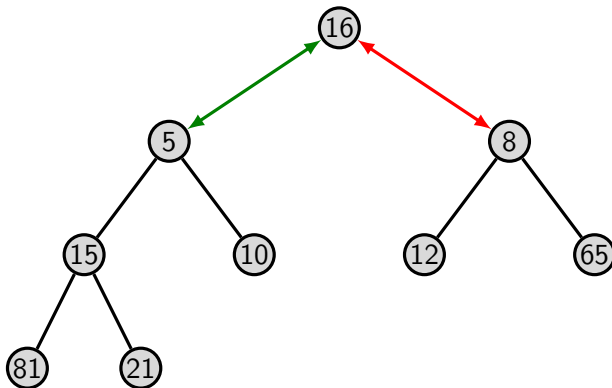
Silme Örneği

Kökteki 4 değerini silme



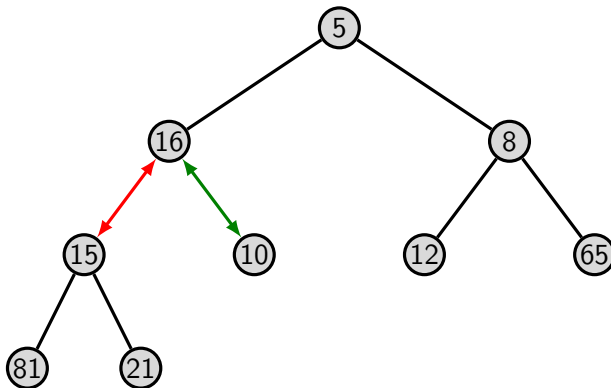
Silme Örneği

Kökteki 4 değerini silme



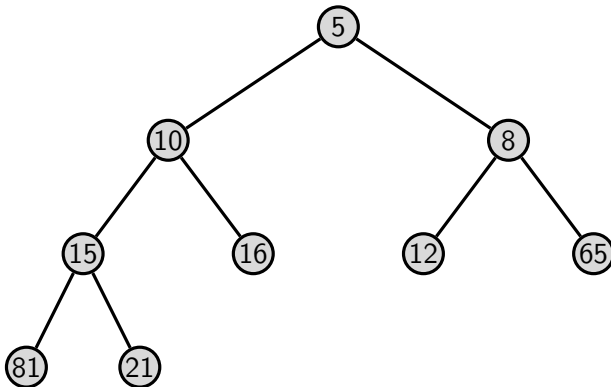
Silme Örneği

Kökteki 4 değerini silme



Silme Örneği

Kökteki 4 değerini silme



2-3 Ağaçları Tanımı

Tanım

Bütün yaprakları aynı seviyede olan, 2-düğümü veya 3-düğümleri barındıran arama ağacıdır. 2-düğümünün tek anahtar değeri(K) ve 2 tane çocuğu mevcuttur. 3-düğümünün iki anahtar değeri(K_1, K_2) ve 3 çocuğu mevcuttur($K_1 < K_2$).

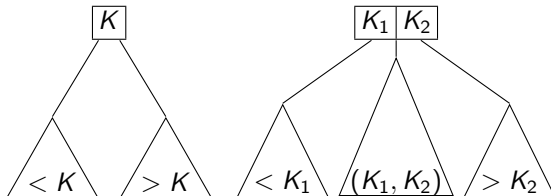


Figure: 2-düğümü ve 3-düğümü

2-3 Ağaçları Örnek

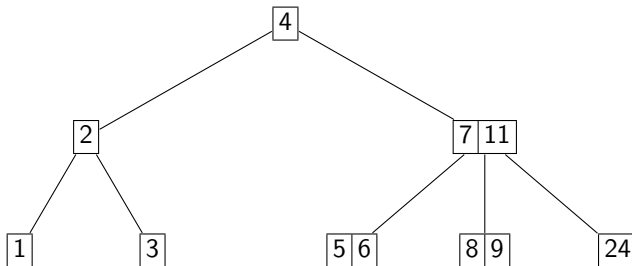


Figure: 2-3 ağacı örneği

Arama İşlemi

ARA(Aranan ,Düğüm)

Eğer Düğüm boş ise

Bulunamadı

Eğer Düğüm 2–Düğümü ise

Eğer Düğüm.Değer=Aranan ise

Bulundu

Eğer Aranan < Düğüm. Değer ise

ARA(Aranan , Dügüm. Sol)

Aksi Takdirde

ARA(Aranan , Dügüm. Sağ)

Aksi Halde(3-Düğümü)

Eğer $Düğüm.Değer1 = Aranan$ veya $Düğüm.Değer2 = Aranan$ ise

Bulundu

Eğer Aranan < Düğüm. Değer1

ARA(Aranan , Düğüm. Sol)

Eğer Aranan>Düğüm.Değer1 ve Aranan<Düğüm.Değer2

ARA(Aranan , Düğüm. Orta)

Aksi Halde

ARA(Aranan . Dügüm. Sol)

Ekleme

- Ekleme her zaman yapraklara yapılır
- Eğer ekleme sırasında düğüm 4-düğümü olursa
 - Ortadaki değer bir üst seviyeye çıkarılır
 - Kalan değerler üst seviyeye çıkan değerın solu ve sağındaki çocukları olurlar
 - Bir üst seviyedeki düğüm de 4-düğümü olmuşsa aynı işlemler tekrarlanır

Ekleme Örneği

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.

Ekleme Örneği

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.

3

Ekleme Örneği: 6

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.

3	6
---	---

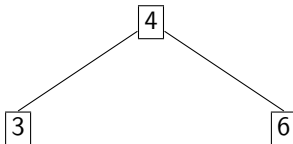
Ekleme Örneği: 4

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.

3	4	6
---	---	---

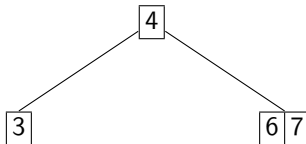
Ekleme Örneği: 4

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.



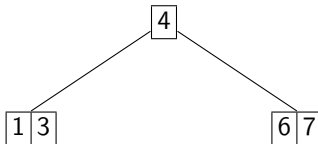
Ekleme Örneği: 7

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.



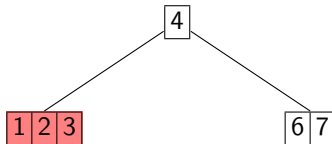
Ekleme Örneği: 1

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.



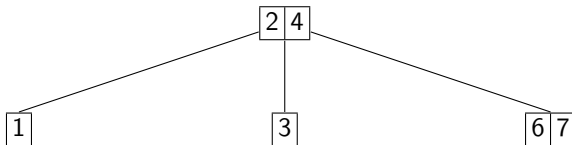
Ekleme Örneği: 2

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.



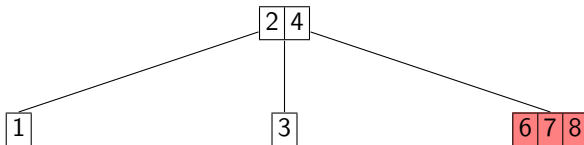
Ekleme Örneği: 2

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.



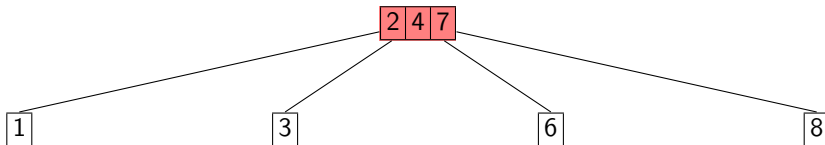
Ekleme Örneği: 8

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.



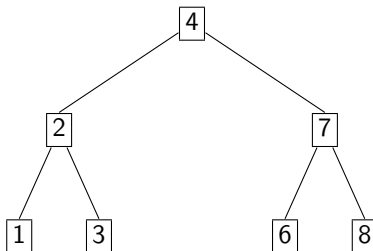
Ekleme Örneği: 8

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.



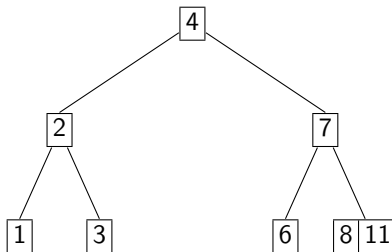
Ekleme Örneği: 8

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.



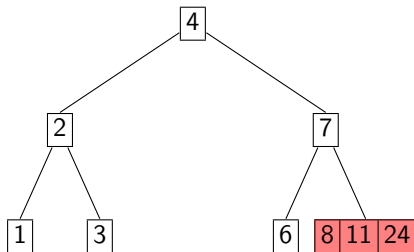
Ekleme Örneği: 11

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.



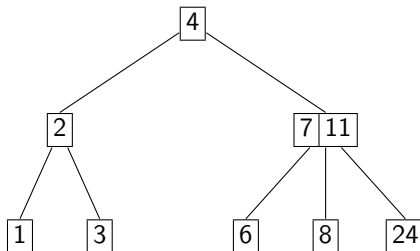
Ekleme Örneği: 24

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.



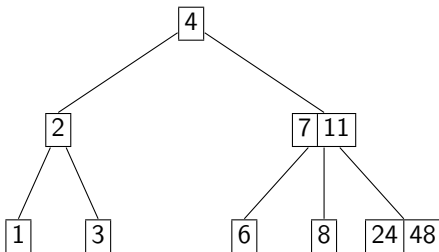
Ekleme Örneği: 24

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.



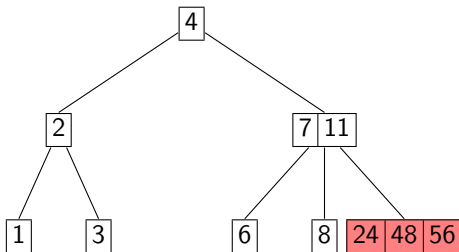
Ekleme Örneği: 48

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.



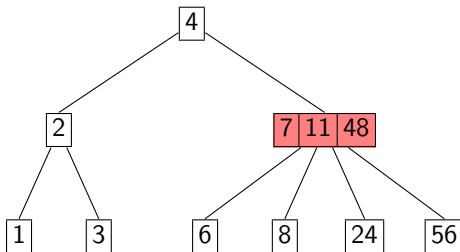
Ekleme Örneği: 56

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.



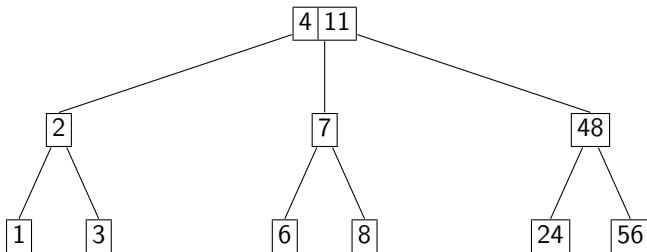
Ekleme Örneği: 56

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.



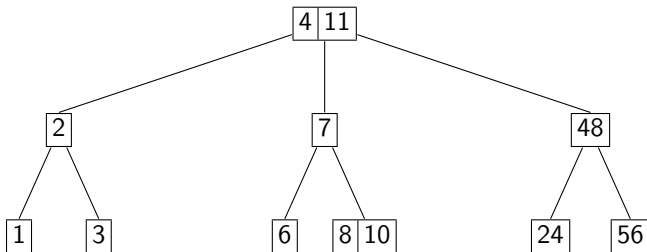
Ekleme Örneği: 56

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.



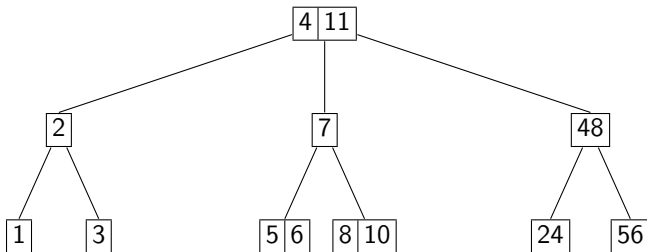
Ekleme Örneği: 10

3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.



Ekleme Örneği: 5

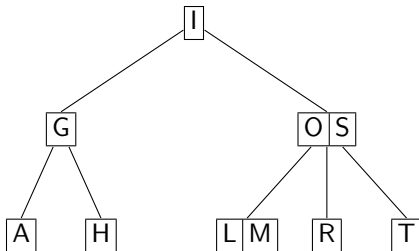
3, 6, 4, 7, 1, 2, 8, 11, 24, 48, 56, 10, 5 değerlerini sırasıyla 2-3 ağacına ekleyelim.



A L G O R I T H M S kelimesinin harflerini sırasıyla 2-3 ağacına ekleyin.

Soru

A L G O R I T H M S kelimesinin harflerini sırasıyla 2-3 ağacına ekleyin.



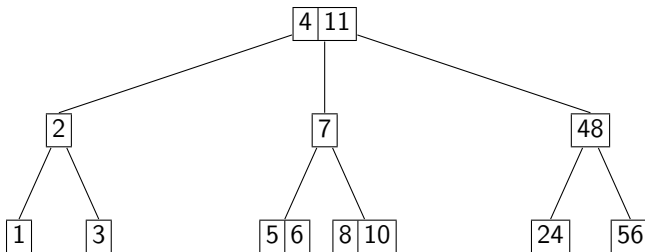
Silme

Silinecek değer yaprakta ise doğrudan silinir, aksi halde öncel(predecessor) veya ardıl(successor) elemanı ile yer değiştirip silinir. Silinen düğümlerde hiç eleman kalmaması durumunda iki farklı durum ortaya çıkar

- Eğer kardeş düğümden ödünç alınabilecek bir değer varsa döndürme yapılır. Kardeş düğüm ve ebeveyn bir üçlü oluşturur. Ortadaki değer ebeveyn olur, küçük olan sol, büyük olan sağ çocuğu oluşturur.
- Aksi halde(kardeşten ödünç alınamazsa) ebeveyn ve kardeş birleştirilip bir alt seviyeye indirilir. Bu işlem köke doğru iletilir.

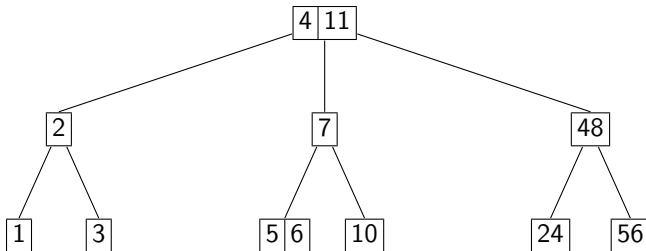
Silme: 8

8 değerini silelim.



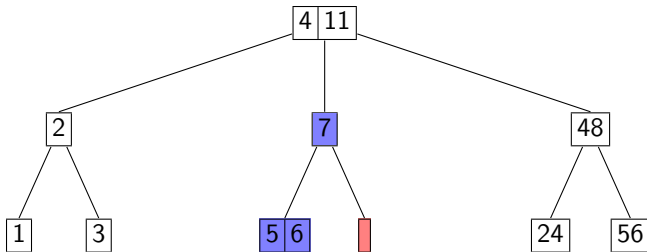
Silme: 10

10 değerini silelim.



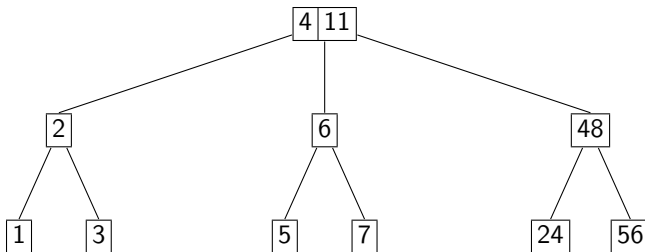
Silme: 10

10 değerini silelim. Durum 1: kardeşten ödünç al ve döndür



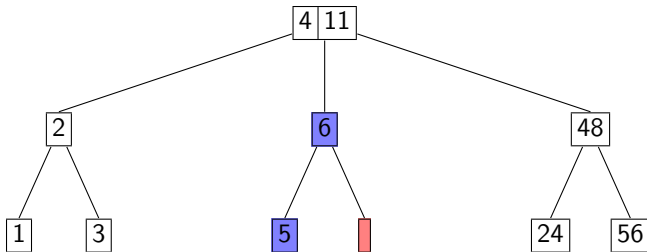
Silme: 7

7 değerini silelim.



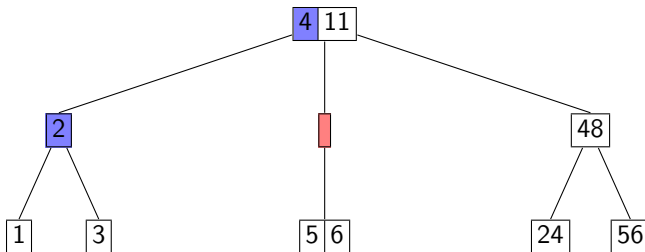
Silme: 7

7 değerini silelim. Durum 2: ebeveyn aşağı inip birleşir



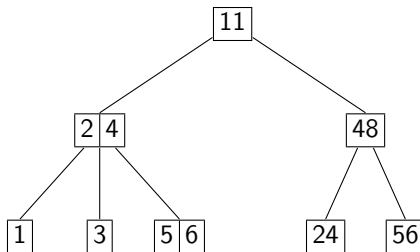
Silme: 7

7 değerini silelim. Durum 2



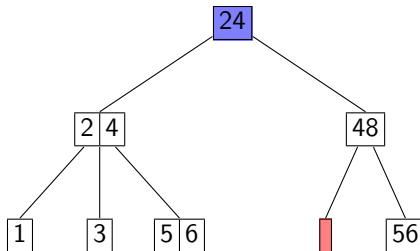
Silme: 11

11 değerini silelim.



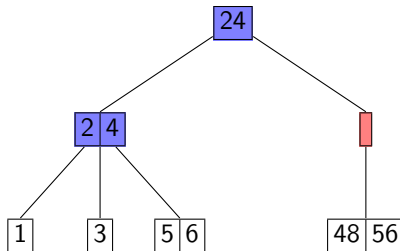
Silme: 11

11 değerini silelim. Durum 2



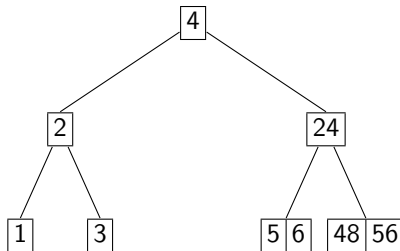
Silme: 11

11 değerini silelim. Durum 1



Silme: 11

11 değerini silelim.



B-Ağaçları

- 2-3 ağaçlarının genel halidir
- Dallanma faktörü(branching factor) değeri vardır(B)
- $B \leq \text{Çocuk sayısı} < 2B$
- $B-1 \leq \text{Düğümdeki eleman sayısı} < 2B-1$
- Düğümdeki elemanlar sıralıdır
- Tüm yapraklar aynı seviyededir
- 2-3 ağacı için B değeri 2'dir

B-Ağaçları

$B=3$ değeri için

- Çocuk sayısı 3,4,5 olabilir(kök düğüm hariç)
- Eleman sayısı 2,3,4 olabilir(kök düğüm hariç)

B-Ağaçları

B=3 değeri için

- Çocuk sayısı 3,4,5 olabilir(kök düğüm hariç)
- Eleman sayısı 2,3,4 olabilir(kök düğüm hariç)

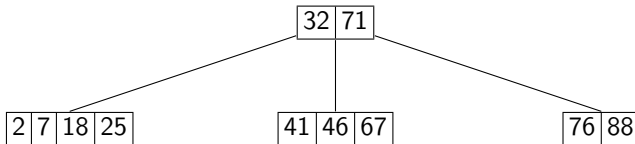


Figure: B=3 için örnek ağaç

Ekleme

- Ekleme işlemi 2-3 ağaçlarındakine benzer yapılır
- Eleman sayısı en büyük değeri geçerse orta değer bur üstte çıkartılır

B-Ağaçları Ekleme: A

A, L, G, O, R, I, T, H, M, S değerlerini B-Ağacına ekleyelim.

A

B-Ağaçları Ekleme: L

A, L, G, O, R, I, T, H, M, S değerlerini B-Ağacına ekleyelim.

A	L
---	---

B-Ağaçları Ekleme: G

A, L, G, O, R, I, T, H, M, S değerlerini B-Ağacına ekleyelim.

A	G	L
---	---	---

B-Ağaçları Ekleme: O

A, L, G, O, R, I, T, H, M, S değerlerini B-Ağacına ekleyelim.

A	G	L	O
---	---	---	---

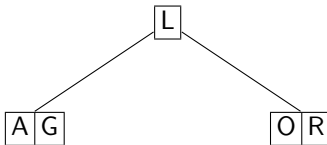
B-Ağaçları Ekleme: R

A, L, G, Ø, R, I, T, H, M, S değerlerini B-Ağacına ekleyelim.

A	G	L	Ø	R
---	---	---	---	---

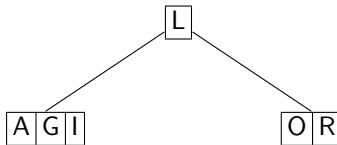
B-Ağaçları Ekleme: R

A, L, G, O, R, I, T, H, M, S değerlerini B-Ağacına ekleyelim.



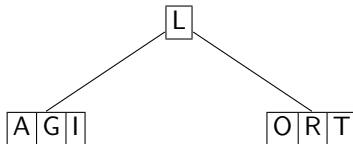
B-Ağaçları Ekleme: I

A, L, G, O, R, t, T, H, M, S değerlerini B-Ağacına ekleyelim.



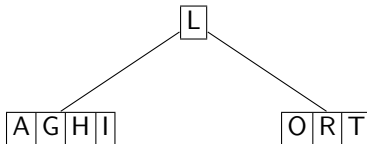
B-Ağaçları Ekleme: T

A, L, G, O, R, t, T, H, M, S değerlerini B-Ağacına ekleyelim.



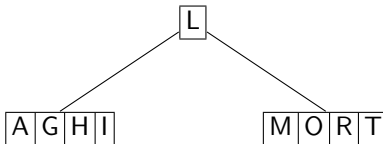
B-Ağaçları Ekleme: H

A, L, G, O, R, t, T, H, M, S değerlerini B-Ağacına ekleyelim.



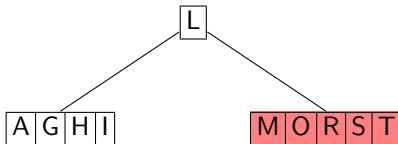
B-Ağaçları Ekleme: M

A, L, G, O, R, İ, T, H, M, S değerlerini B-Ağacına ekleyelim.



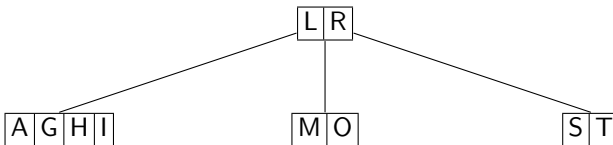
B-Ağaçları Ekleme: S

A, L, G, O, R, I, T, H, M, S değerlerini B-Ağacına ekleyelim.



B-Ağaçları Ekleme: S

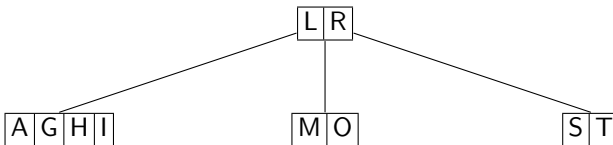
A, L, G, O, R, İ, T, H, M, S değerlerini B-Ağacına ekleyelim.



- Silme işlemi 2-3 ağaçlarındaki gibidir
- Ağacı düzenleme(döndürme/birleştirme) en az düğüm sayısının altına düşülürse gerçekleşir

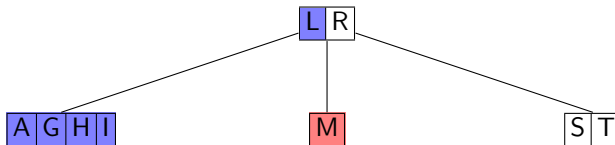
B-Ağaçları Silme: O

O değerini silelim.



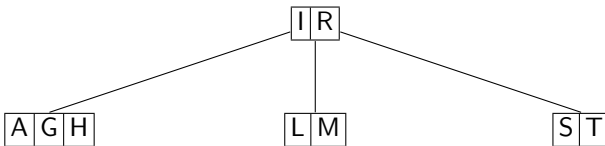
B-Ağaçları Silme: O

O değerini silelim. Durum 1



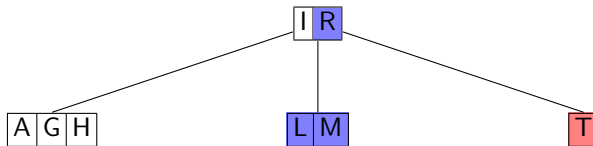
B-Ağaçları Silme: S

S değerini silelim.



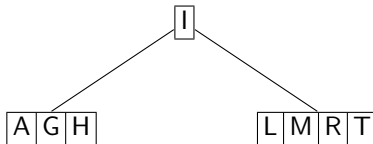
B-Ağaçları Silme: S

S değerini silelim. Durum 2



B-Ağaçları Silme: S

S değerini silelim.



Ağaç İşlemleri Zaman Karmaşıklıkları(En kötü durum)

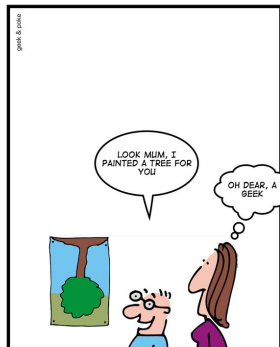
Table: Temem işlemlerin zaman karmaşıklıkları

Data Structure	Search	Insert	Delete
BST	$O(n)$	$O(n)$	$O(n)$
AVL	$O(\log n)$	$O(\log n)$	$O(\log n)$
B-Tree	$O(\log n)$	$O(\log n)$	$O(\log n)$
Heap	$O(n)$	$O(\log n)$	$O(\log n)$

Görselleştirme

- İkili Yığın(Binary Heap) için
<http://www.cs.usfca.edu/galles/visualization/Heap.html>
- 2-3 ağaçları ve B-ağaçları için
<http://www.cs.usfca.edu/galles/visualization/BTree.html>, B=2 için
Max. Degree 3, B=3 için Max. Degree 5 seçilmeli

LITTLE GEEKS



THE ROOT IS UP, PERIOD!