# **İkili Arama Ağacı (20,15,10,30,40)**

**Özellikleri**

* İlk önce kök düğüm oluşturulur
* Sadece sol ve sağ çocuk eklenebilir
* Sol çocuklar daima parent’tan küçük olmalıdır
* Sağ çocuklar daima parent’tan büyük olmalıdır
* Her Eklenen düğüm leaf(yaprak) durumundadır.(En alt)

Ekle(20) > İlk düğüm olan kök 20 doğrudan eklenir.

20

Ekle(15) > 15 kök düğüm olan 20 den küçük olduğu için sola eklenir.

20

15

Ekle(15) > 10 kök düğüm ve sol ilk düğüm(15) ten küçük olduğu için 15 in soluna eklenir.

20

15

10

Ekle(30) > 30 kök düğümden büyük olduğu için sağına eklenir.

10

15

20

30

Ekle(40) > 40 kök düğüm ve onun sağındaki 30 dan büyük olduğundan 30 un sağına eklenir.

20

15

10

30

40

# **AVL Ağacı (20,15,10,30,40)**

**Dengeli İkili Ağaç**

* İkili arama ağacının dezavantajı, yüksekliğin N-1 kadar olabilmesidir.
* Bunun manası: Ekleme ve silme ve diğer birçoğu işlem gerçekleştirilirken karmaşıklığın en kötü durumda O(N) olmasıdır.
* İstenen özellik ağacın yüksekliğinin küçük olmasıdır.
* N tane düğümü olan ikili ağacın yüksekliği en az O(log N) ‘ dir.
* Bu nedenle, amaç ikili arama ağacının yüksekliğini O(log N) olarak tutabilmektir.
* Bu tür ağaçlara dengeli ikili ağaçlar denilir. Örnekler AVL ağaçlar, kırmızı-siyah ağaçlar

Ekle(20) > İlk eklenen kök düğümdür.

20

**1**

Ekle(15) > 15 kök düğüm olan 20 den küçük olduğu için sola eklenir.

20

**2**

20

**1**

Ekle(10) > Ağaca yeni eklenen 10 ardından ağacın dengelenmesi için 15 kök yapılıp sağına 20 ve soluna 10 eklenir.

15

**2**

10

**1**

20

**1**

Ekle(30) > Eklenen 30 değerinde dengeleme durumu olmadığı için ağacın sağına eklenir. 20 ağacın sağındadır ve 30 20 den büyüktür bu nedenle 20 nin sağına eklenir.

15

**3**

10

**1**

20

**2**

30

**1**

Ekle(40) > 40 değeri eklendiğinde ağaçı tekrar dengelemek için 30 ile 20 yer değiştirir ve 40 ağacın sağındaki 30 un sağına 20 ise 39 un soluna eklenir.

15

**3**

10

**1**

30

**2**

40

**1**

20

**1**

# **Heap (20,15,10,30,40)**

**Öbek**

* Öbekler, eklemeleri soldan sağa doğru yapılan ve son seviyesi dışındaki tüm seviyeleri dolu olan bir ikili ağaç çeşididir.
* Öbeklerin ana özellikleri her ebeveyn değerin çocuklarından daha küçük / büyük değere sahip olmasıdır.
* Ebeveynin indisi: (indis - 1) / 2
* Sol çocuğun indisi: 2 \* indis + 1
* Sağ çocuğun indisi: 2 \* indis + 2

**MİN HEAP**

Ekle(20) > İlk düğüm olan kök 20 doğrudan eklenir.

20

Ekle(15) > 15 bir üst düğüm olan 20 den küçüktür ve yer değişimi yapılır yeni düğüm alt düğüm boş olduğundan sola eklenir.

15

20

Ekle(10) > 10 değeri üst düğüm olan 15 den küçüktür ve yer değişimi yapılır, yeni düğüm sol düğüm dolu olduğu için sağa eklenir.

10

15

20

Ekle(30) > 30 değeri öbekteki en büyük değerden büyük olduğu için öbeğin en sol altına eklenir.

30

20

10

15

Ekle(40) > 40 değeri öbekteki en büyük değer olan 30 dan büyüktür fakat öbekteki henüz 20 değerinin sağı boştur ve 40, 20’nin sağına eklenir.

10

20

30

15

40

## **İkili Arama Ağacı Sıralama**

* Bir dizi oluşturulur (ağacın boyutu kadar)
* Kökten başlanarak soldaki tüm düğümler (her bir düğüme gelindiğinde o düğüm kök görevi yani ebeveyn görür)
* Soldaki tüm düğümler gezildikten sonra sağdaki düğümler gezilir.
* Her bir düğüm için bu aşamalar tekrar edilir.

**Sırasıyla >**

* Ağacın solunun sonuna kadar gidilir ve en küçük değer buradakidir.
* Bir üst düğüme dönülür ve sağ taraf kontrol edilir varsa seçilir.
* Bu aşamalar ağaçtaki tüm düğümler için gerçekleştirilir.
* Sonuç olarak bir dizi içerisine ağaçtaki elemanların küçükten büyüğe sıralanmış hali aktarılır.

20

15

10

30

40

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 10 | 15 | 20 | 30 | 40 |

## **AVL Ağacı Sıralama**

* Bir dizi oluşturulur (ağacın boyutu kadar)
* AVL ağaçlarında bir düğümün içerisindeki küçükten büyüğe dizilim şöyledir;
* Çocuk Sol(10) < Ebeveyn(15) < Çocuk Sağ(30)

15

**2**

10

**1**

30

**1**

**Sırasıyla >**

* Kökten başlanır ve ağacın sol sonuna dek gidilir, sol son değer en küçük değerdir.
* Sol değerin ebeveyni ondan büyük bir değerdir ve o seçilir.
* Ebeveynin sağı da ebeveynden büyüktür ve sağı seçilir.
* Yukarı doğru bu adımlar gerçekleştirilir.

15

**3**

10

**1**

30

**2**

40

**1**

20

**1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 10 | 15 | 20 | 30 | 40 |

## **Heap Sıralama**

* Bir dizi oluşturulur (ağacın boyutu kadar)
* Min Heap te kök en küçük değerdir.
* O anki düğümün sağı solundan küçüktür.

10

20

15

**Sırasıyla >**

* Kökten başlanır ve önce kök seçilir (kök en küçüktür).
* Sırasıyla Sağ çocuk ve sol çocuk seçilir.
* Bu işlemler her bir düğüm için tekrar edilir.

10

20

30

15

40

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 10 | 15 | 20 | 30 | 40 |