

AVL Ağaçları: Motivasyon

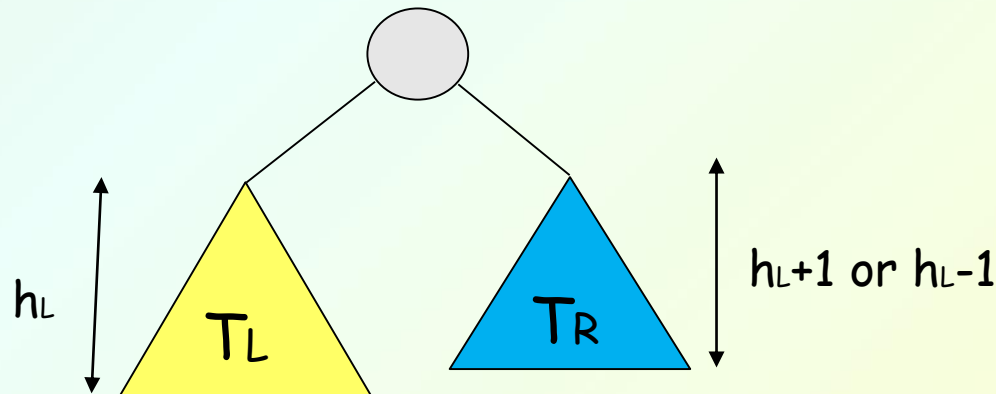
- İkili arama ağaçları üzerindeki tartışmalarımızı hatırlarsak
 - Ağacın yüksekliği ekleme sırasına bağlıdır.
 - Ö.g., 1, 2, 3, 4, 5, 6, 7 sayılarını boş bir ağaca eklersek
 - Problem: Dengenin bozulması - Ağaç bağlantılı liste şeklinde bozulur.
 - Tüm operasyonların karmaşıklığı $O(h)$ olduğunda, $\log N \leq h \leq N-1$, İkili arama ağacı operasyonlarının en kötü çalışma zamanı $O(N)$ olur.
- Soru: Ekleme sırasına bakmadan BST nin yüksekliği $\log(n)$ olabilir mi? Bir başka deyişle, BST'yi dengeli bir şekilde tutabilir miyiz.

Yükseklik Dengeli Ağaçlar

- BST operasyonlarını daha kısa sürede gerçekleştirmek için pek çok BST dengeleme algoritması vardır.
 - Adelson-Velskii ve Landis (AVL) ağaçları (1962)
 - Splay ağaçları (1978)
 - B-ağacı ve diğer çok yönlü arama ağaçları (1972)
 - Red-Black ağaçları (1972)
 - Ayrıca Simetrik İkili B-Ağaçları(Symmetric Binary B-Trees) şeklinde de bilinir

AVL Ağaçları: Tanım

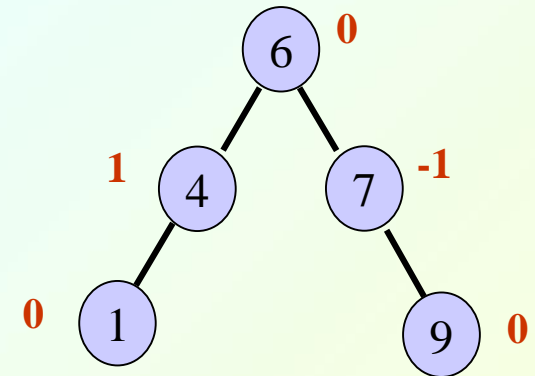
1. Tüm boş ağaç AVL ağacıdır.
2. Eğer T boş olmayan T_L ve T_R şeklinde sol ve sağ alt ağaçları olan ikili arama ağacı ise, T ancak ve ancak aşağıdaki şartları sağlarsa AVL ağacı şeklinde isimlendirilir.
 1. T_L ve T_R AVL ağaçları ise
 2. h_L ve h_R T_L ve T_R nin yükseklikleri olmak üzere $|h_L - h_R| \leq 1$ olmak zorundadır.



AVL Ağaçları

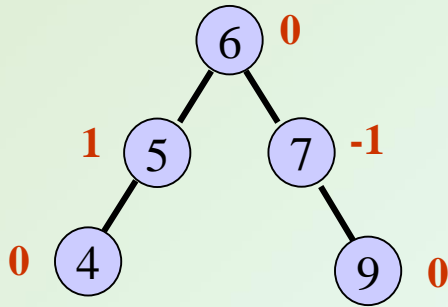
- AVL ağaçları dengeli ikili arama ağaçlarıdır.
- Bir düğümdeki denge faktörü =
 $\text{yükseklik}(\text{sol altağaç}) - \text{yükseklik}(\text{sağ altağaç})$
- AVL ağaçlarında balans faktörü sadece $-1, 0, 1$ olabilir.
- Her bir düğümün sol ve sağ alt ağaçlarının yükseklikleri arasındaki fark en fazla 1 olabilir.

AVL Ağacı

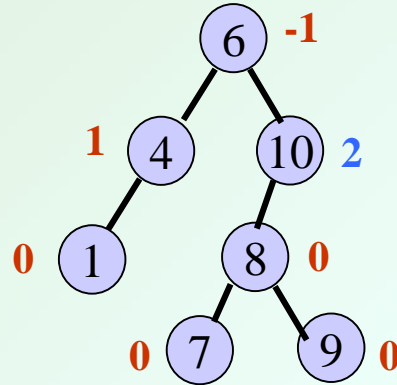


Kırmızı numaralar balans faktörüdür.

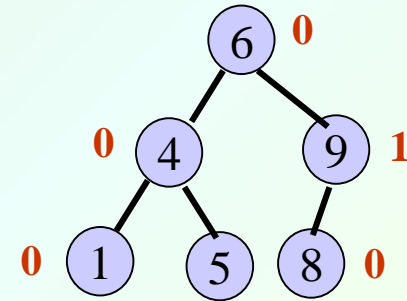
AVL Ağaçları: Örnekler



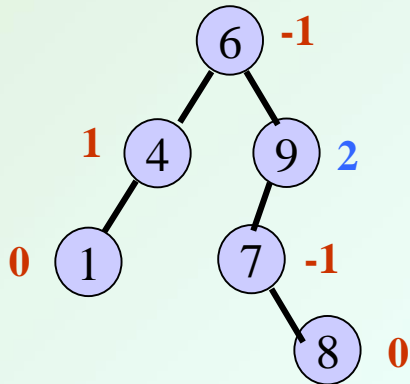
AVL Ağacı



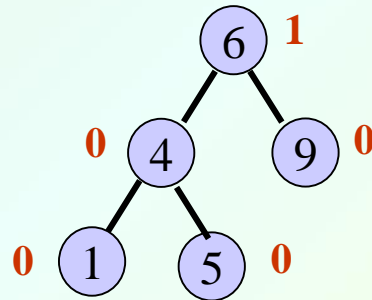
AVL Ağacı Değildir



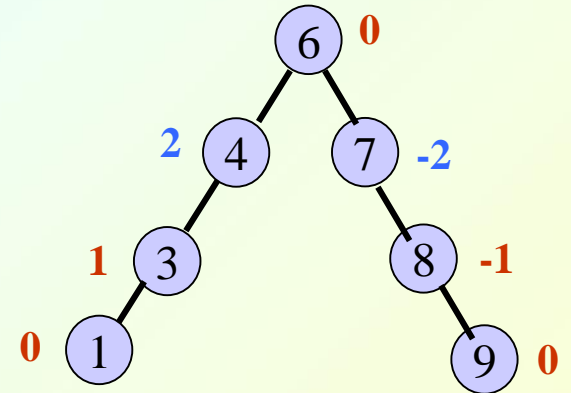
AVL Ağacı



AVL Ağacı Değildir



AVL Ağacı

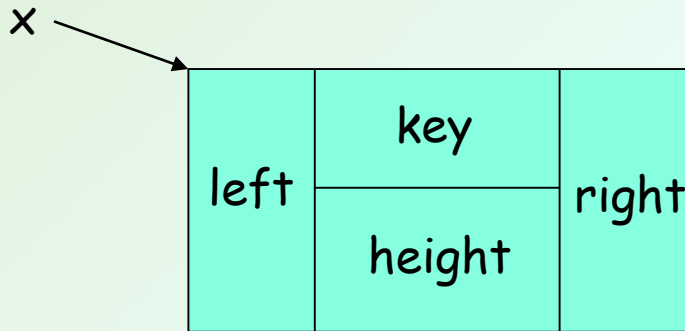


AVL Ağacı Değildir

Kırmızı numaralar balans faktörüdür.

AVL Ağacı: Gerçekleştirim

- AVL ağacının gerçekleştirimi için her x düğümünün yüksekliği kaydedilir.



```
class AVLDugumu{  
    int deger;  
    int yukseklik;  
    AVLTDugumu sol;  
    AVLDugumu sag;  
}
```

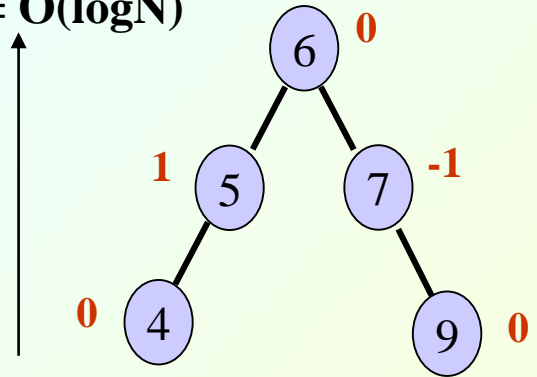
- x 'in balans faktörü = x 'in sol alt ağacının yüksekliği - x 'in sağ alt ağacının yüksekliği
- AVL ağaçlarında, "bf" sadece $\{-1, 0, 1\}$ değerlerini alabilir.

AVL Ağaçları

- N düğümlü bir AVL ağacının yüksekliği daima $O(\log N)$ dir.
- Peki nasıl?

Yükseklik = $O(\log N)$

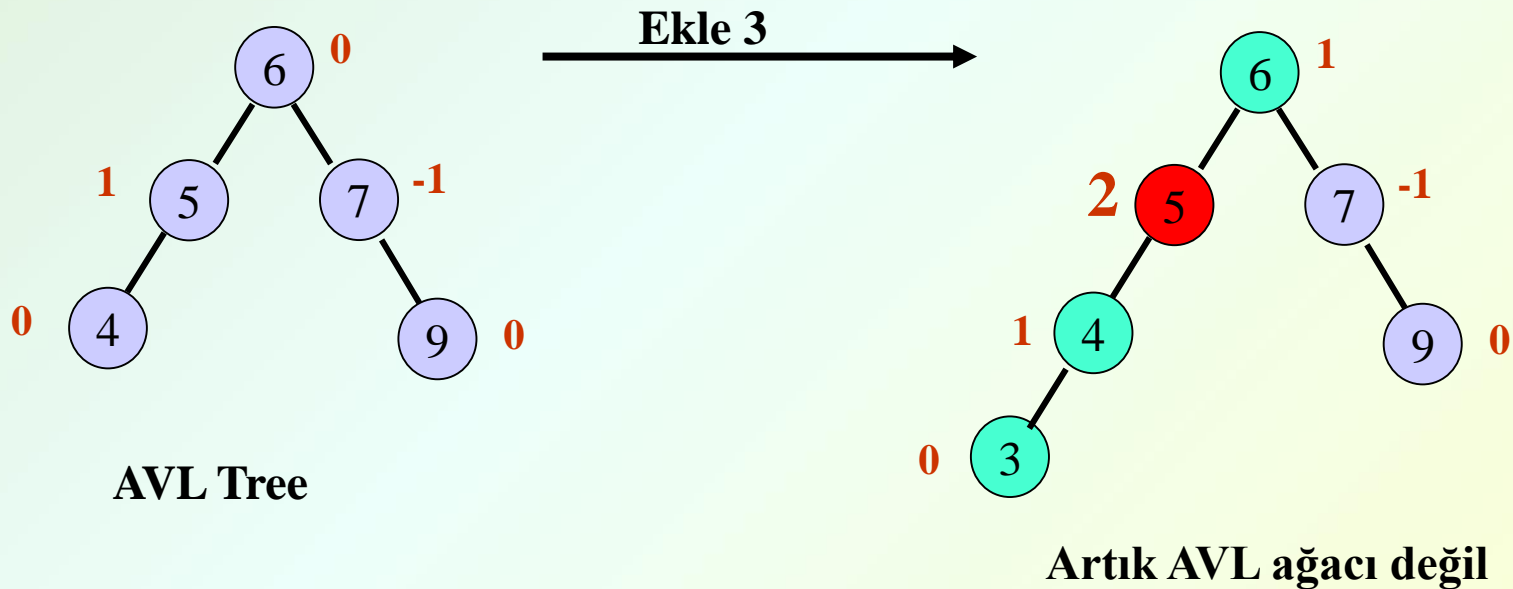
AVL Ağacı



Kırmızı numaralar balans faktörüdür.

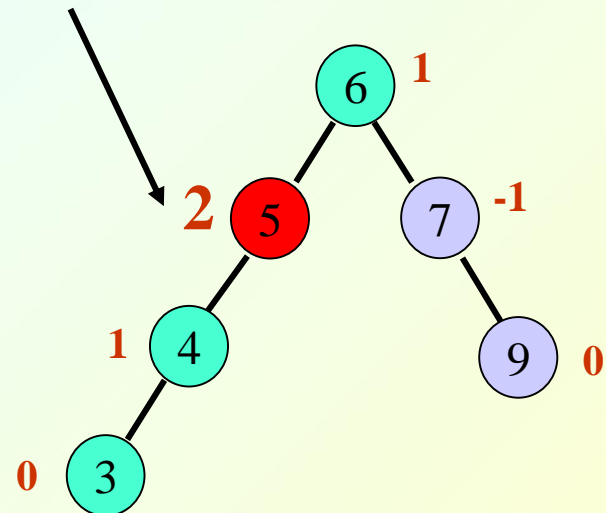
AVL Ağaçları: Güzel ve Çirkin

- Güzel:
 - Arama süresi $O(h) = O(\log N)$
- Çirkin
 - Ekleme ve silme işlemleri ağacın dengesiz olmasına neden olabilir.

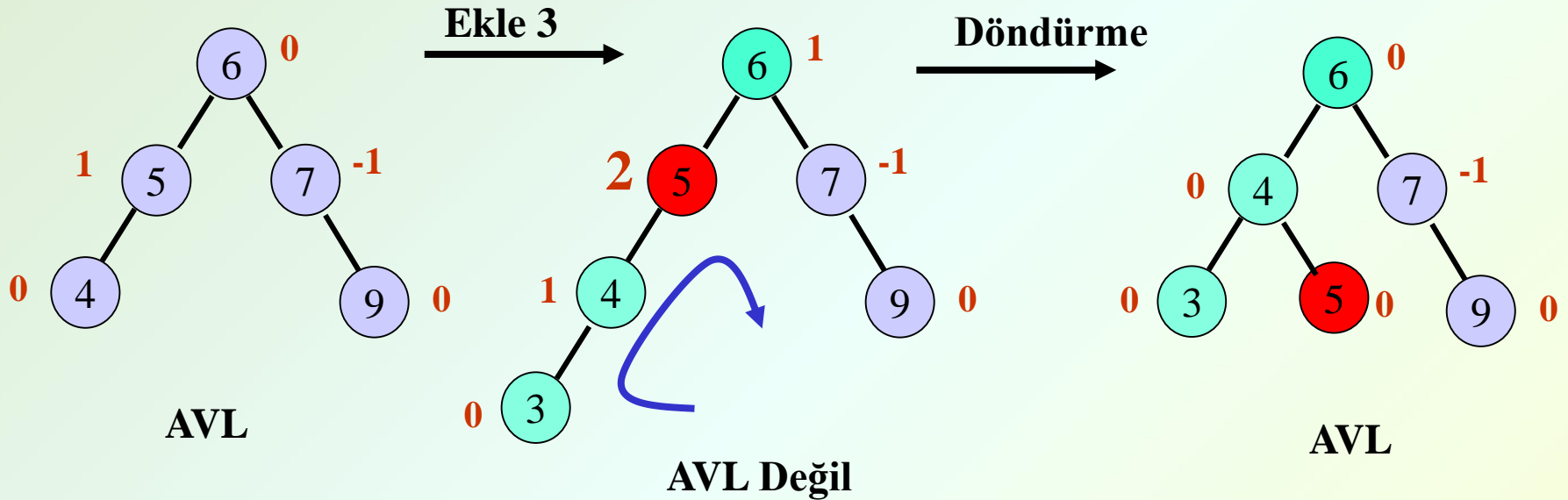


AVL Ağacında Dengenin Sağlanması

- **Problem:** Ekleme işlemi bazı durumlarda **ekleme noktasına göre kök olan bölgelerde** balans faktörün 2 veya -2 olmasına neden olabilir.
- **Fikir:** Yeni düğümü ekledikten sonra
 1. Balans faktörü düzelterek köke doğru çık.
 2. Eğer düğümün balans faktörü 2 veya -2 ise ağaç bu düğüm üzerinde döndürülerek düzeltilir.

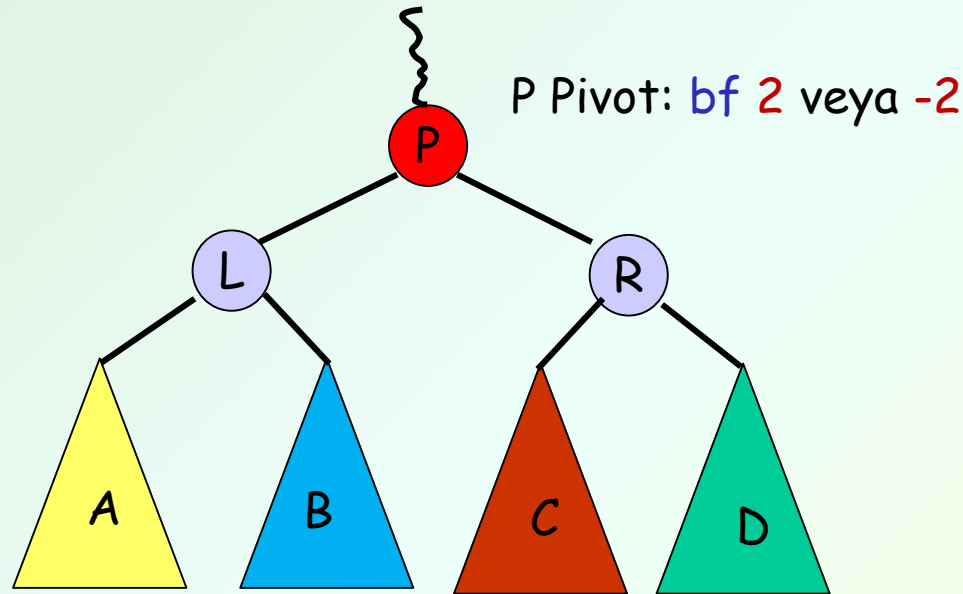


Denge Sağlama: Örnek



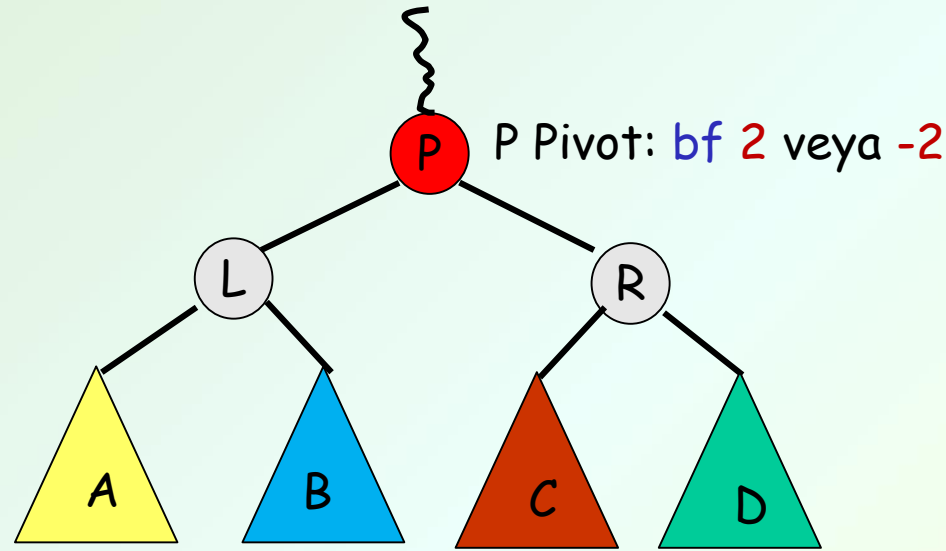
- Yeni düğümü ekledikten sonra
 - Balans faktörü düzelterek köke doğru çık.
 - Eğer düğümün balans faktörü 2 veya -2 ise ağaç bu düğüm üzerinde döndürülerek düzeltilir.

AVL Ağacı - Ekleme (1)



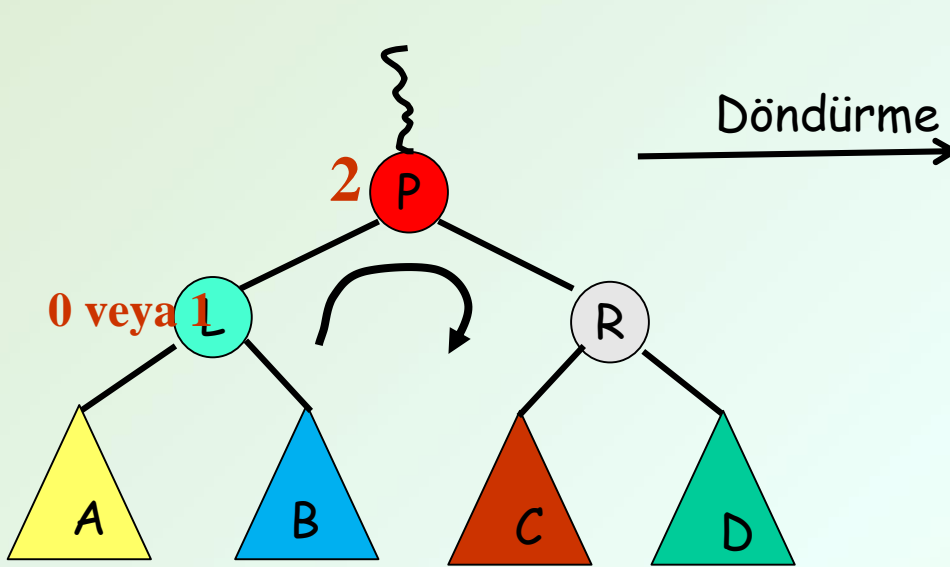
- **P** düğümünün dengeyi bozan düğüm olduğu düşünülürse.
 - **P pivot** düğüm şeklinde isimlendirilir.
 - Eklemeden sonra köke doğru çıkarken bf'nin 2 veya -2 olduğu ilk düğümdür.

AVL Ağacı - Ekleme (2)

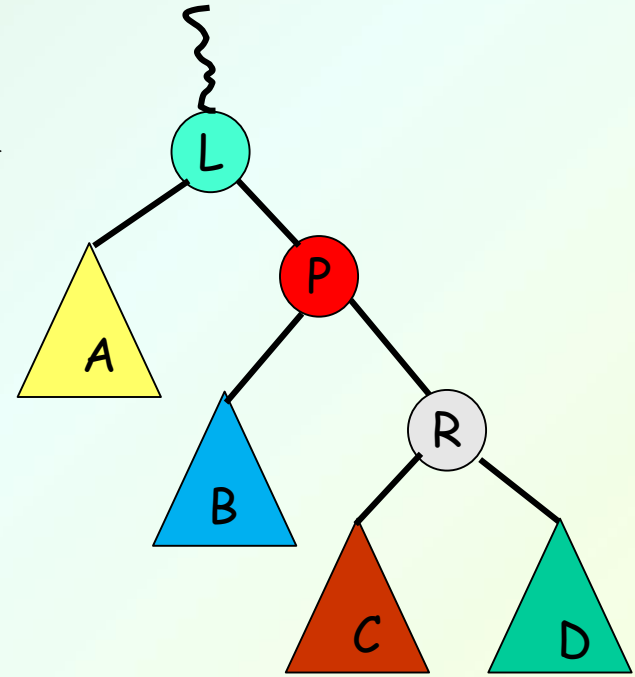


- 4 farklı durum vardır:
 - **Dış Durum** (tek döndürme gerektiren) :
 1. P'nin sol alt ağacının soluna eklendiğinde (LL Dengesizliği).
 2. P'nin sağ alt ağacının sağına eklendiğinde (RR Dengesizliği)
 - **İç Durum** (2 kez döndürme işlemi gerektiren) :
 3. P'nin sol alt ağacının sağına eklendiğinde (RL Dengesizliği)
 4. P'nin sağ alt ağacının soluna eklendiğinde (LR Dengesizliği)

LL Dengesizliği & Düzeltme



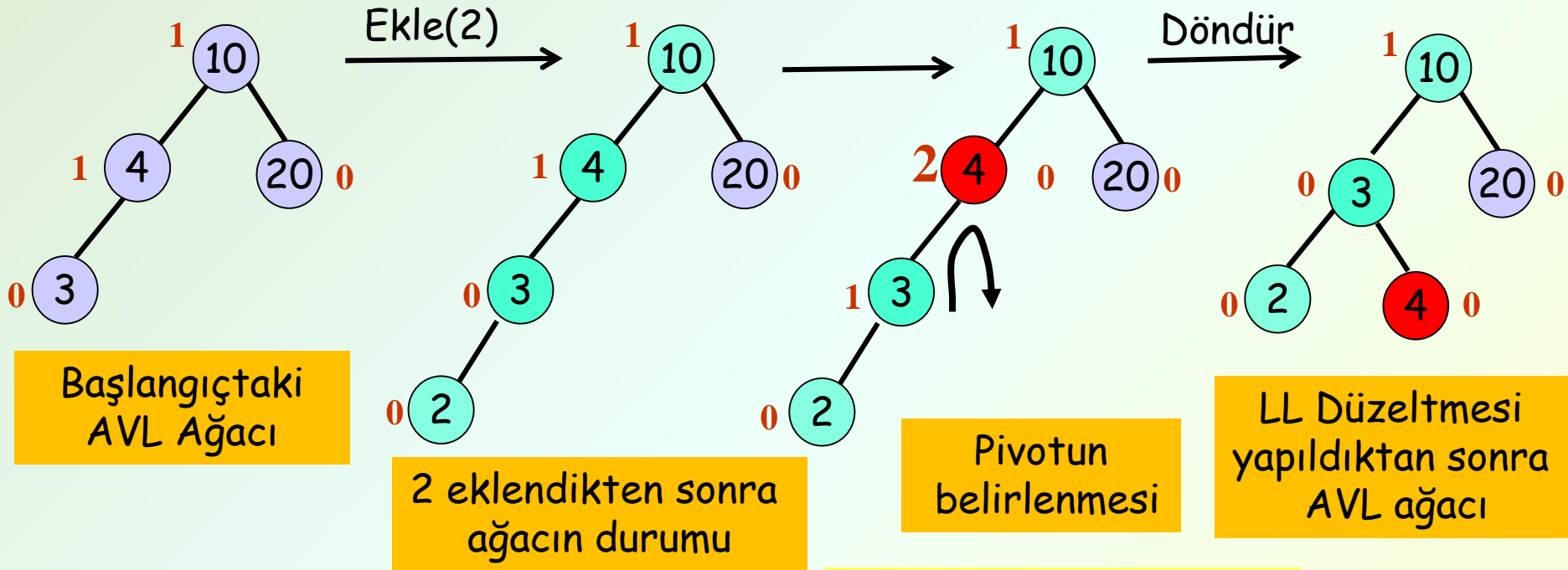
Ekleme işleminden sonra ağaç



LL Düzeltmesinden sonra ağaç

- **LL Dengesizliği:** P'nin sol alt ağacının soluna eklendiğimizde (A alt ağacına)
 - P'nin bfsi 2
 - L'nin bfsi 0 veya 1
- **Düzeltme:** P etrafında sağa doğru dönderme.

LL Dengesizliği Düzeltme Örneği (1)

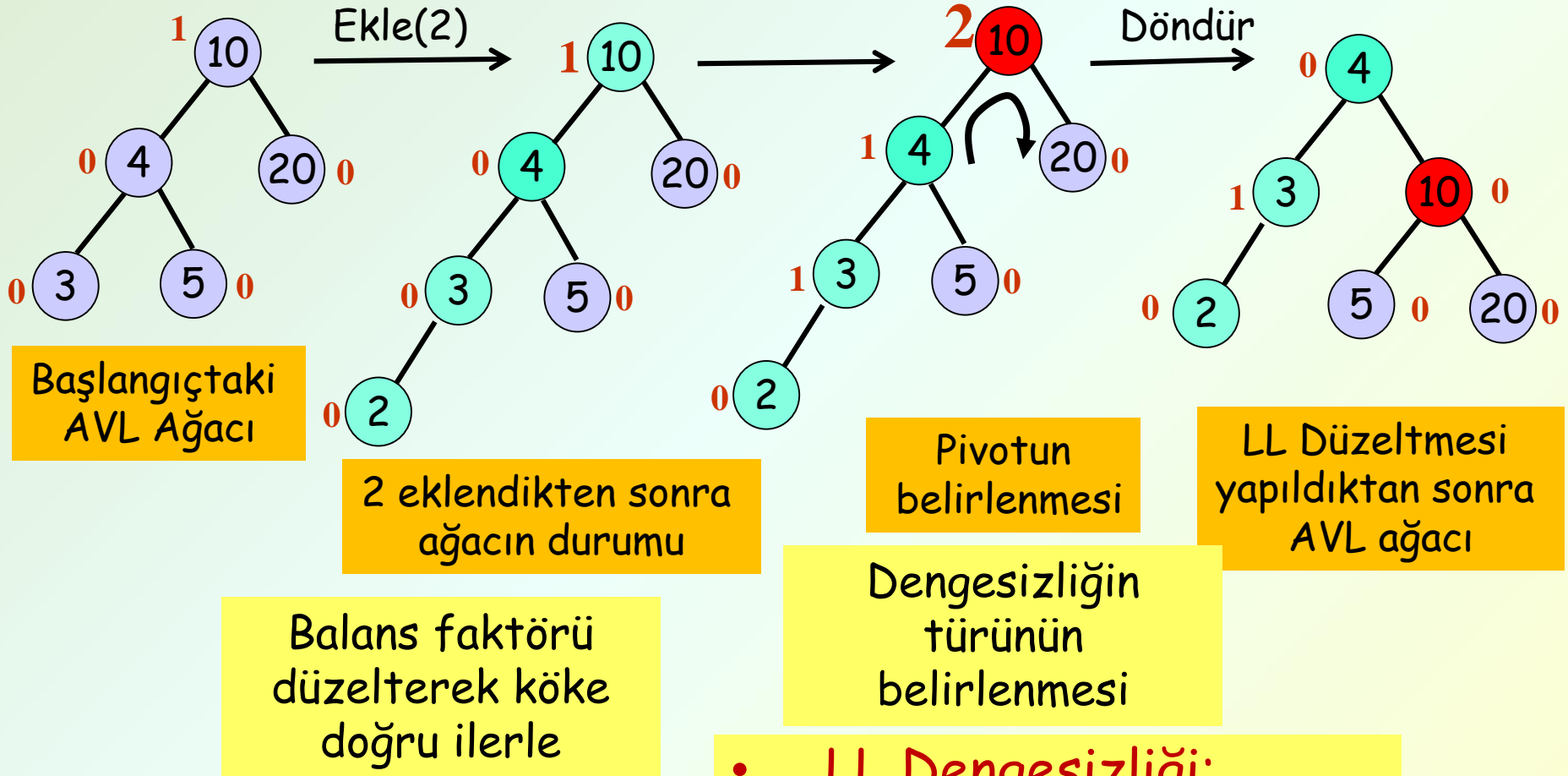


Balans faktörü
düzelterek köke
doğru ilerle

Dengesizliğin türünün
belirlenmesi

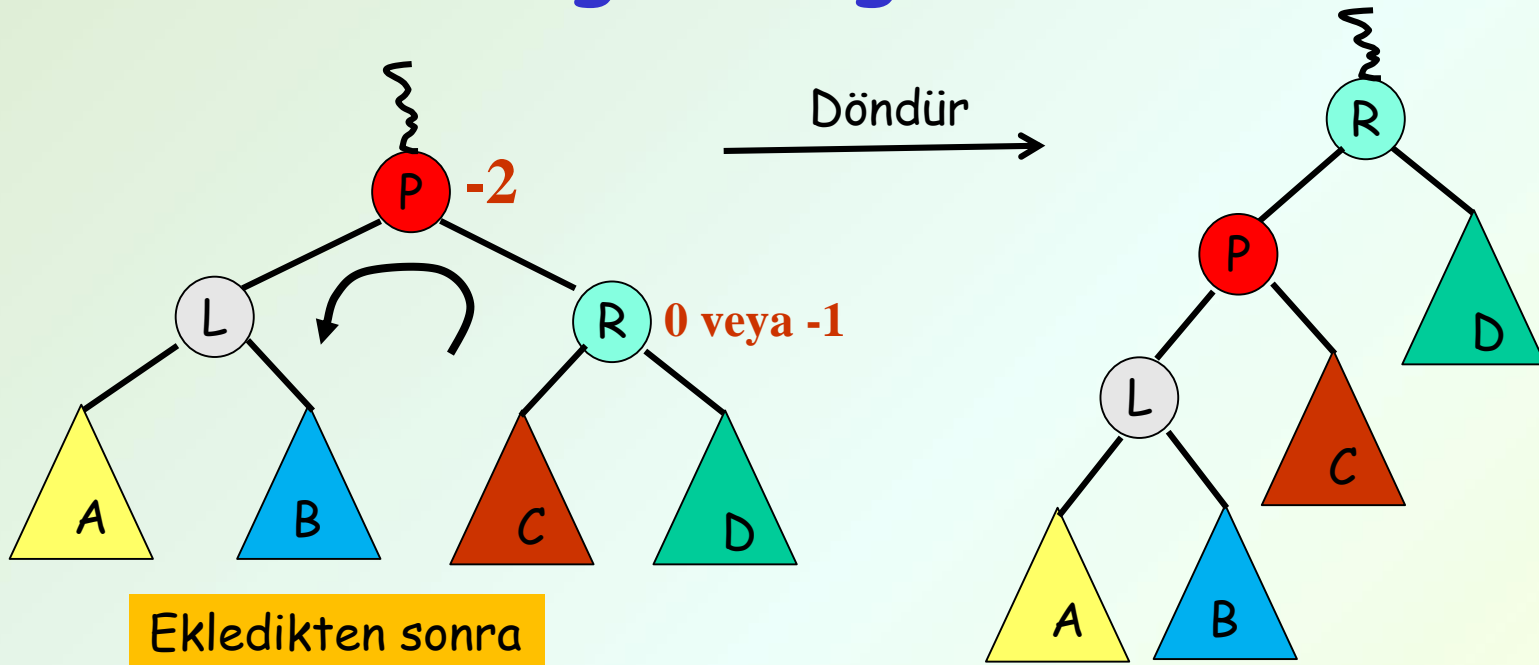
- **LL Dengesizliği:**
 - **P(4)**'ün **bf**si 2
 - **L(3)**'ün **bf**si 0 veya 1

LL Dengesizliği Düzeltme Örneği (2)



- LL Dengesizliği:
 - P(4)'ün bfsi 2
 - L(3)'ün bfsi 0 veya 1

RR Dengesizliği & Düzeltme

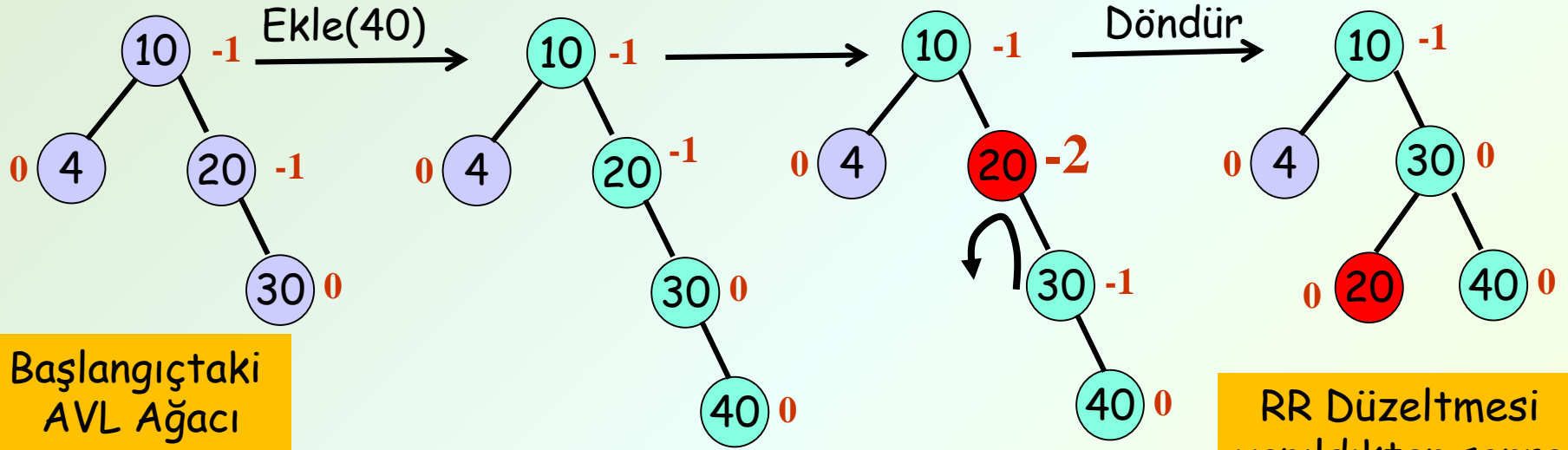


Ekledikten sonra

RR Düzeltmesi yapıldıktan sonra

- **RR Dengesizliği:** P'nin sağ alt ağacının sağına eklendiğinde (D alt ağacına eklendiğinde)
 - $P \rightarrow bf = -2$
 - $R \rightarrow bf = 0 \text{ veya } -1$
- **Düzeltme:** P etrafında sola doğru dönderme

RR Dengesizliği Düzeltme Örneği (1)



40 eklendikten sonra ağacın durumu

Pivotun belirlenmesi

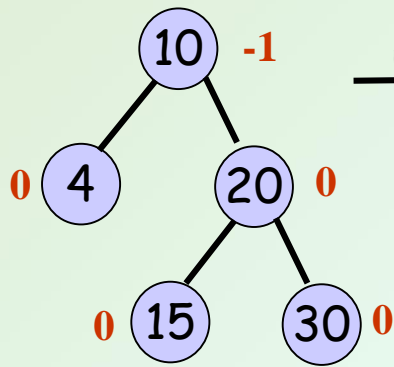
RR Düzeltmesi yapıldıktan sonra AVL ağacı

Balans faktörü düzelterek köke doğru ilerle

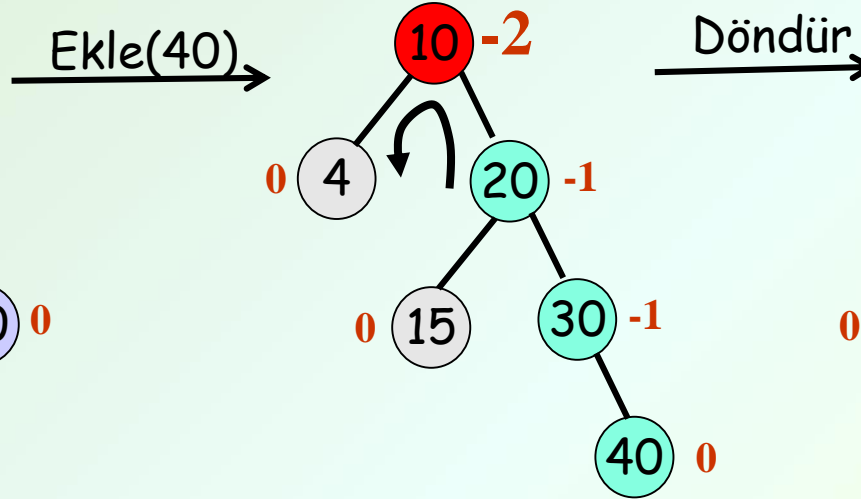
Dengesizliğin türünün belirlenmesi

- **RR Dengesizliği:**
 - $P(20) \rightarrow bf = -2$
 - $R(30) \rightarrow bf = 0$ veya -1

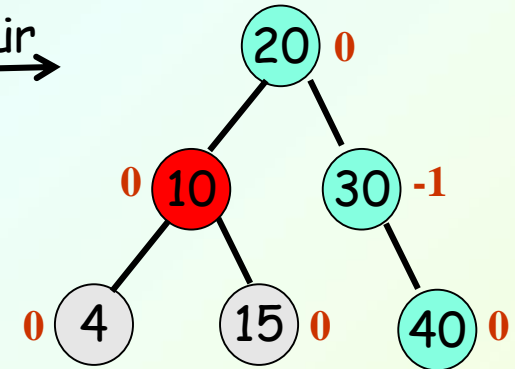
RR Dengesizliği Düzeltme Örneği (2)



Başlangıçtaki AVL Ağacı



40 eklendikten sonra ağacın durumu



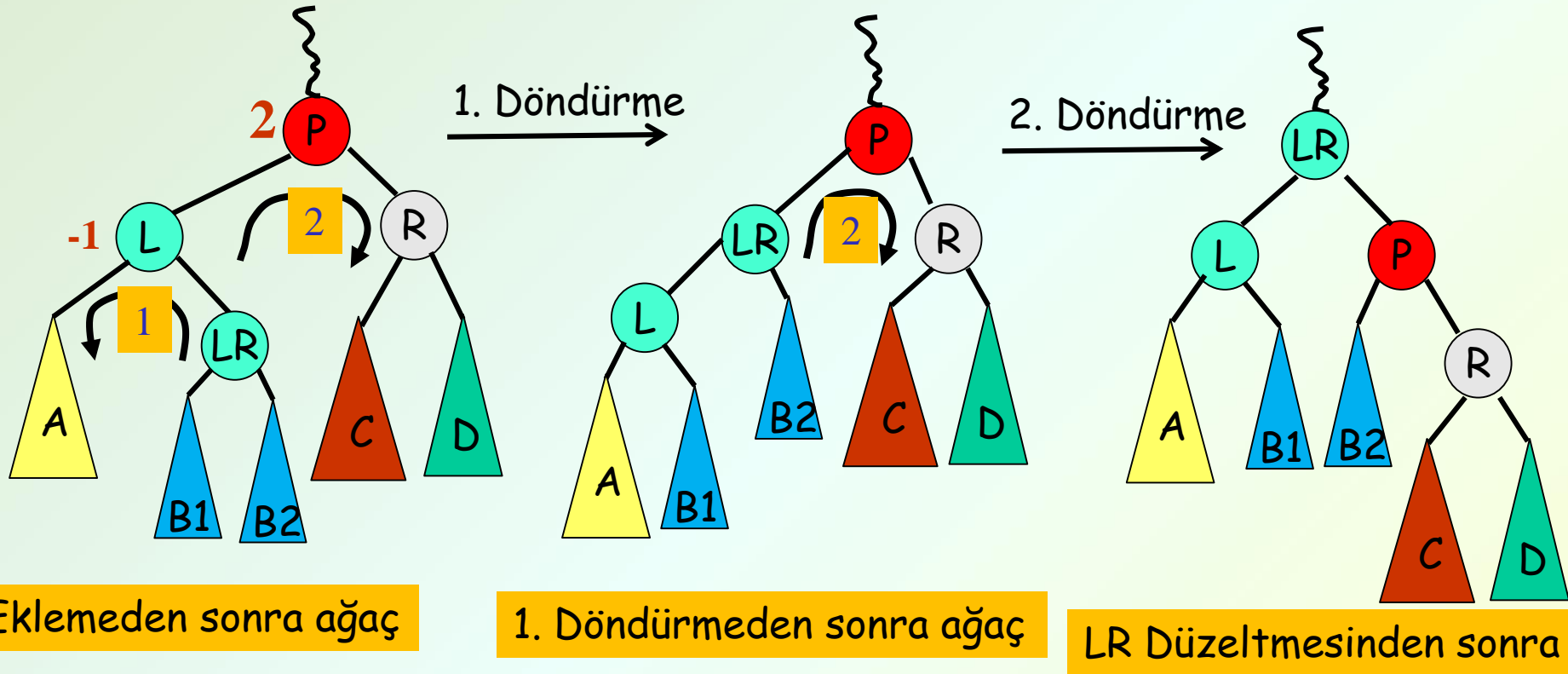
RR Düzeltmesi yapıldıktan sonra AVL ağacı

Balans faktörü düzelterek köke doğru ilerle ve 10'u pivot olarak belirle

Dengesizliğin türünün belirlenmesi

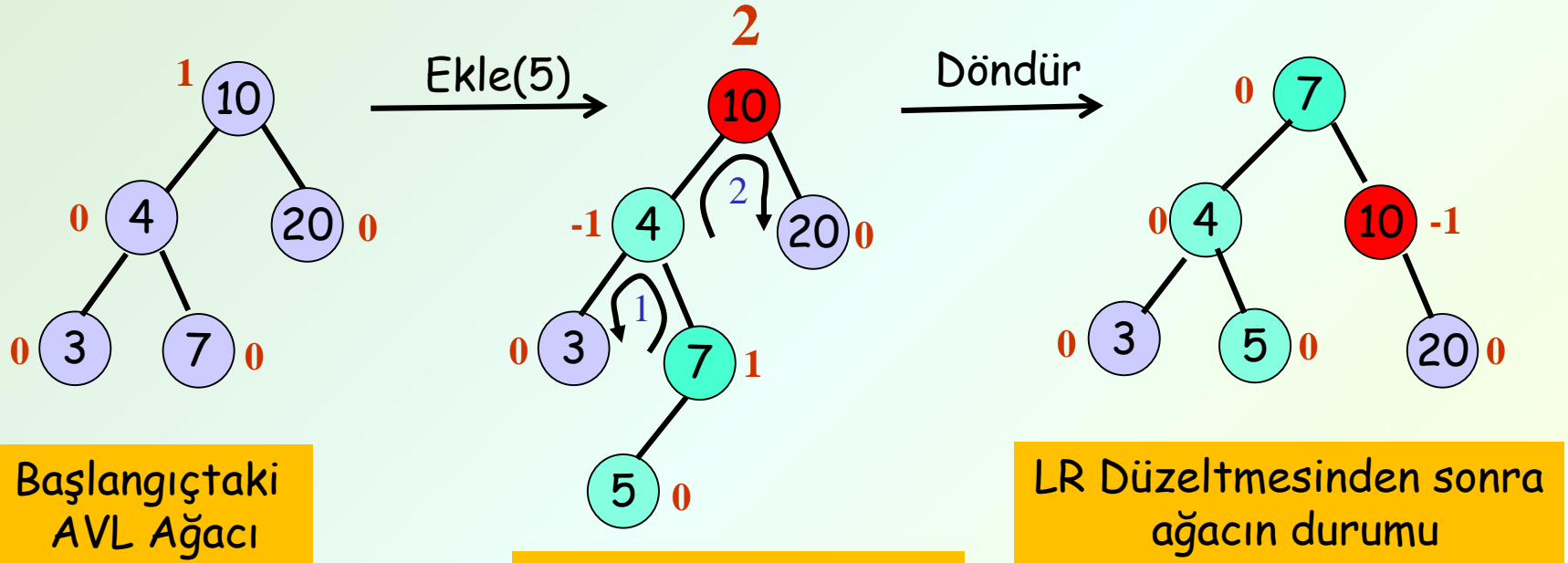
- **RR Dengesizliği:**
 - $P(10) \rightarrow bf = -2$
 - $R(20) \rightarrow bf = 0$ veya -1

LR Dengesizliği & Düzeltme



- **LR Dengesizliği:** P'nin sol alt ağacının sağına eklendiğinde (**LR ağacına**)
 - $P \rightarrow bf = 2$
 - $L \rightarrow bf = -1$
- **Düzeltme:** L & P etrafında 2 kez döndürme

LR Dengesizliği Düzeltme Örneği



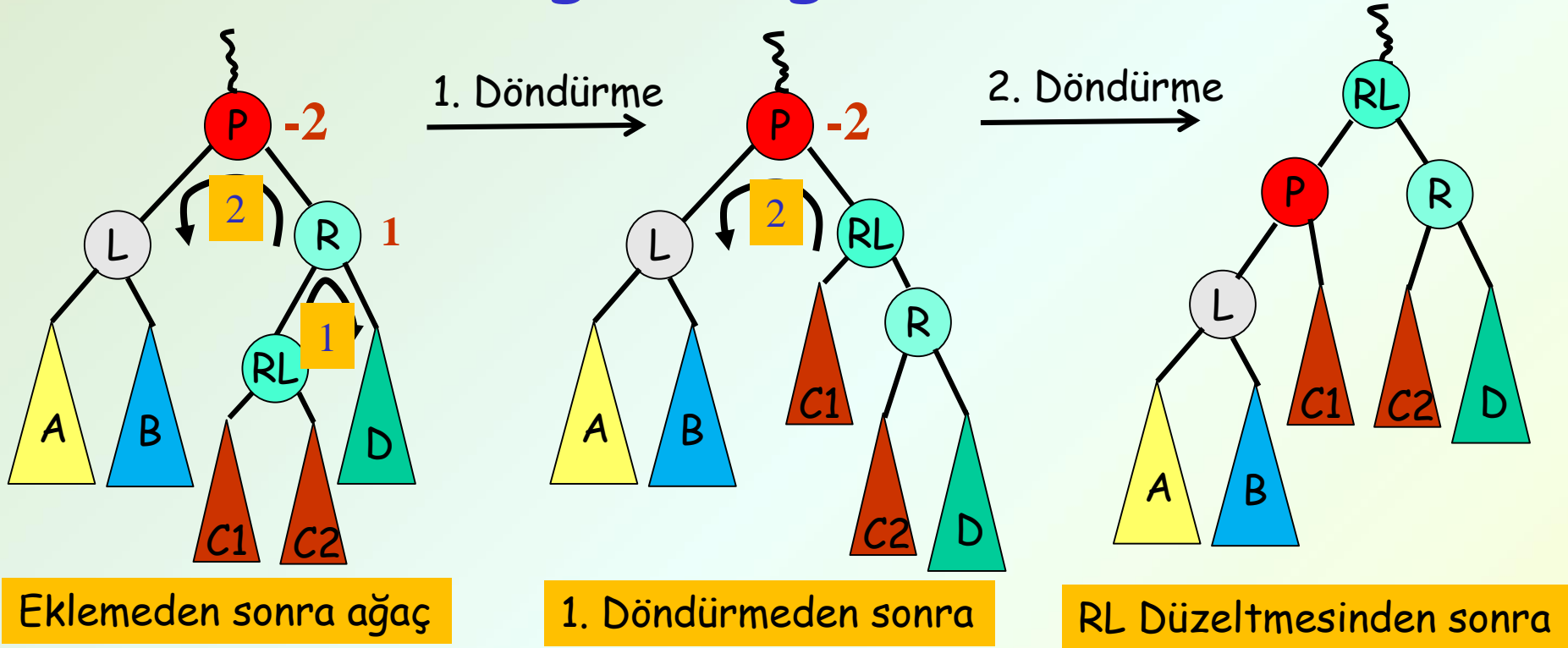
Balans faktörü düzelterek köke doğru ilerle ve 10'u pivot olarak belirle

2 eklendikten sonra ağacın durumu

Dengesizliğin türünün belirlenmesi

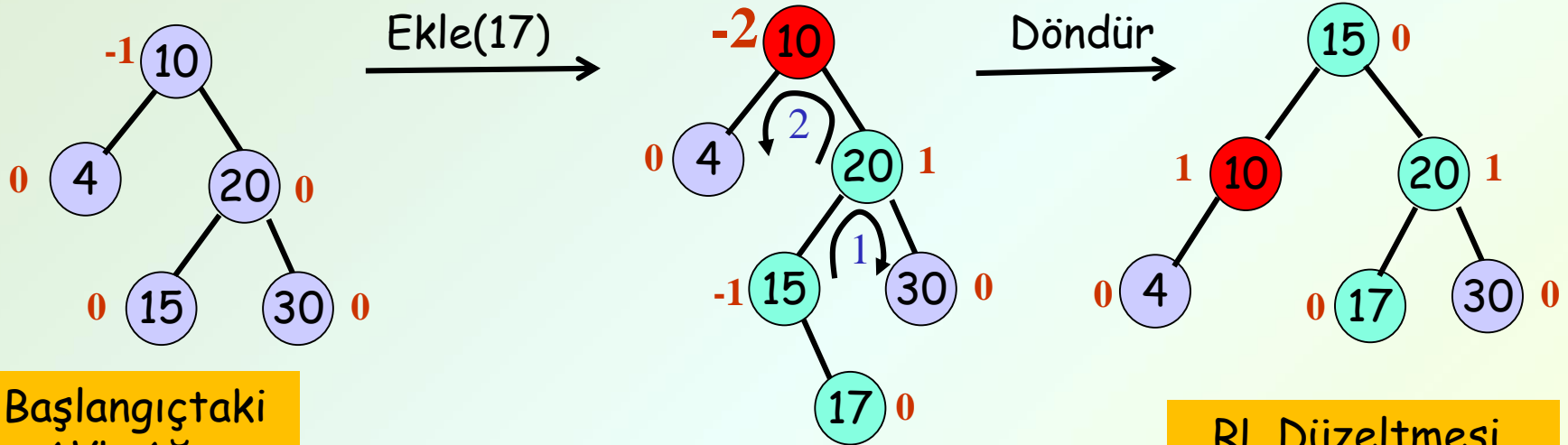
- **LR Dengesizliği:**
 - $P(10) \rightarrow bf = 2$
 - $L(4) \rightarrow bf = -1$

RL Dengesizliği & Düzeltme



- **RL Dengesizliği:** P'nin sağ alt ağacının soluna eklendiğinde (RL alt ağacına)
 - $P \rightarrow bf = -2$
 - $R \rightarrow bf = 1$
- **Düzeltme:** R & P etrafında 2 kez döndürme.

RL Dengesizliği Düzeltme Örneği



Başlangıçtaki
AVL Ağacı

Balans faktörü
düzelterek köke
doğru ilerle ve 10'u
pivot olarak belirle

17 eklendikten sonra
ağacın durumu

Dengesizliğin türünün
belirlenmesi

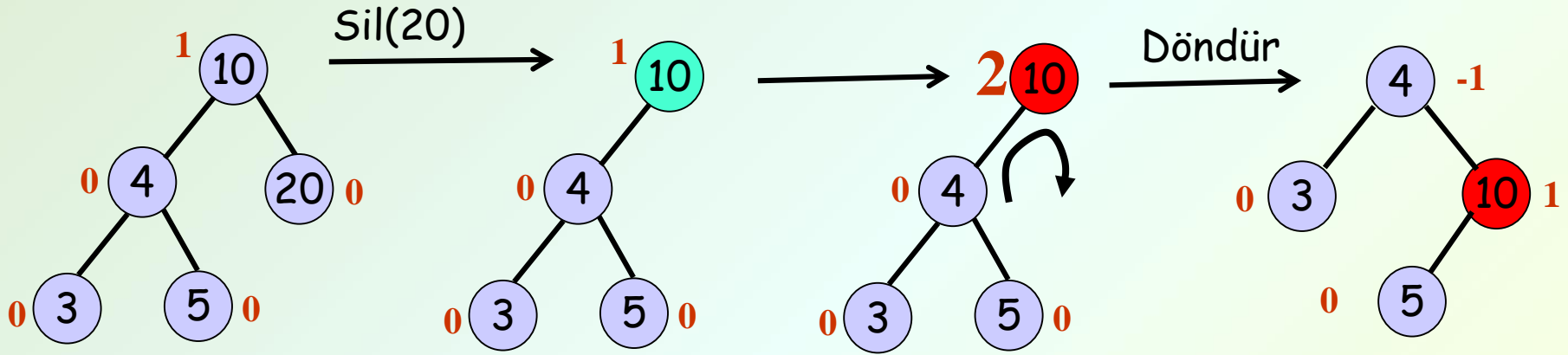
RL Düzeltmesi
yapıldıktan sonra
AVL ağacı

- **RL Dengesizliği:**
 - $P(10) \rightarrow bf = -2$
 - $R(20) \rightarrow bf = 1$

Silme

- Silme işlemi ekleme işlemi ile benzerlik gösterir.
- Öncelikle ikili arama ağacındaki gibi normal silme işlemi yapılır.
- Düğüm silindikten sonra, ağaçta köke doğru ilerleyin ve balans faktörünü güncelleyin.
 - Eğer dengesiz bir düğüm tespit edilirse, uygun döndürme işlemlerini yapın.
 - Bu işlem sırasında birden fazla döndürme işlemi yapılabilir.

Silme Örneği (1)



Başlangıçtaki
AVL Ağacı

20 silindikten sonra
Ağacın durumu

10'un pivot olarak
belirlenmesi

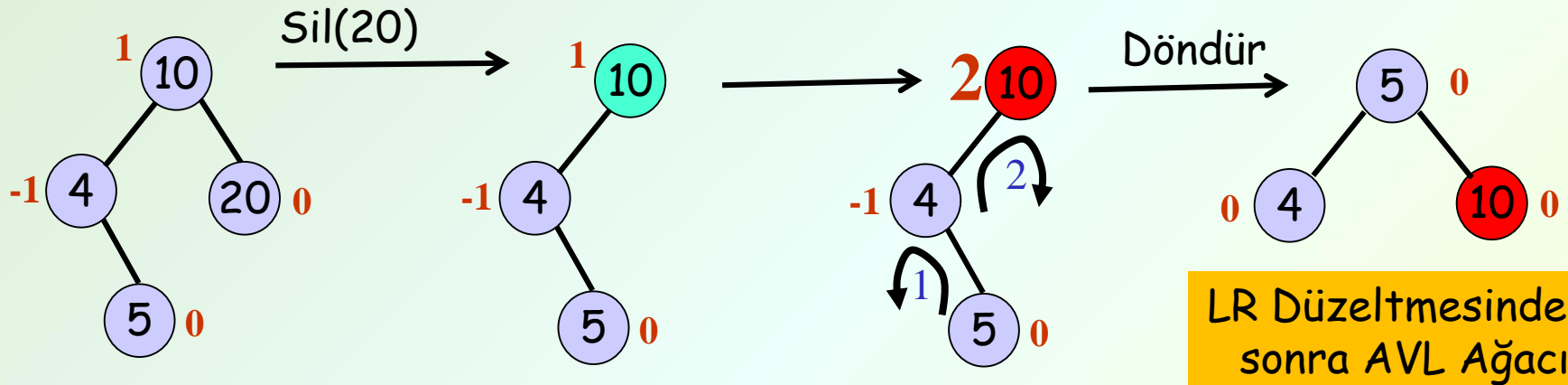
LL Düzeltmesinden
Sonra AVL Ağacı

Balans faktörü
düzelterek köke
doğru ilerle

Dengesizliğin
türünün
belirlenmesi

- LL Dengesizliği:
 - P(10)'ün bfsi 2
 - L(4)'ün bfsi 0 veya 1

Silme Örneği (2)



Başlangıçtaki
AVL Ağacı

20 silindikten sonra
Ağacın durumu

10'un pivot olarak
belirlenmesi

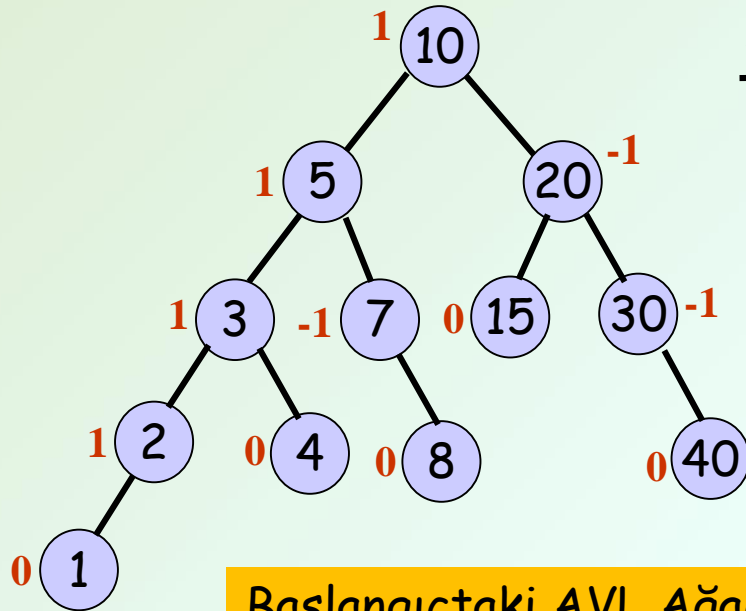
LR Düzeltmesinden
sonra AVL Ağacı

Balans faktörünü
düzelterek köke
doğru ilerle

Dengesizliğin
türünün belirlenmesi

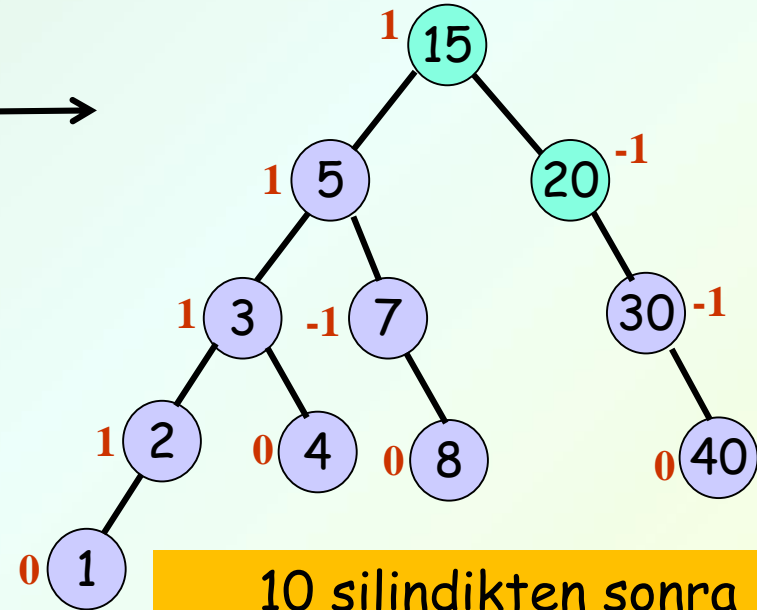
- **LR Dengesizliği:**
 - $P(10) \rightarrow bf = 2$
 - $L(4) \rightarrow bf = -1$

Silme Örneği (3)



Başlangıçtaki AVL Ağacı

Sil(10) →

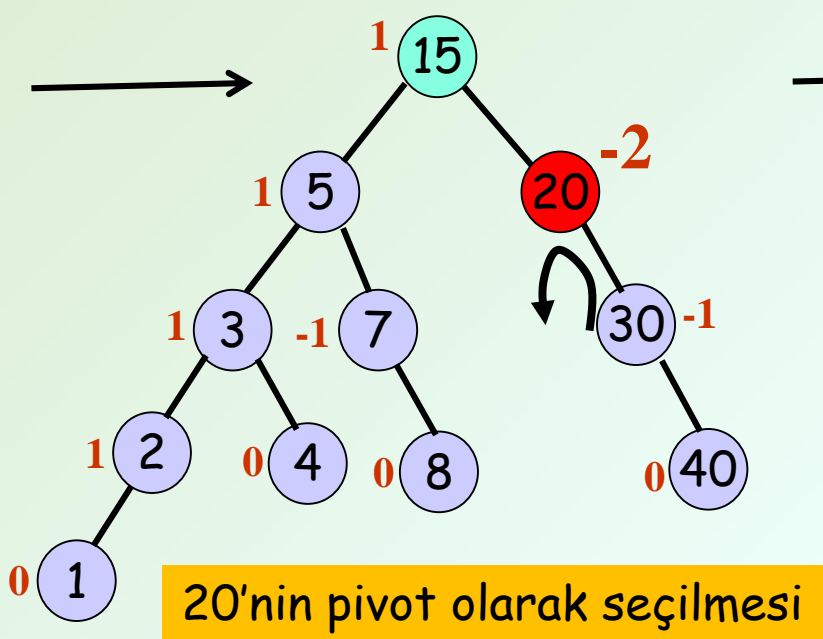


10 silindikten sonra

10 ve 15(sağ alt ağaçtaki en küçük eleman) yer değiştirdi ve 15 silindi.

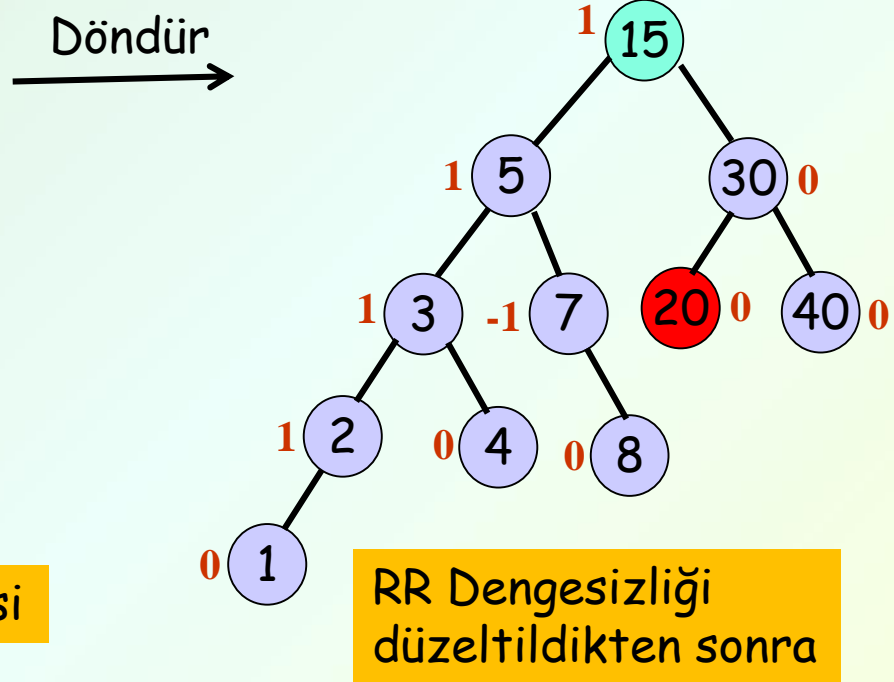
Balans faktörü düzelterek köke doğru ilerle

Silme örneği (3) - devam



Dengesizliğin türünü belirle

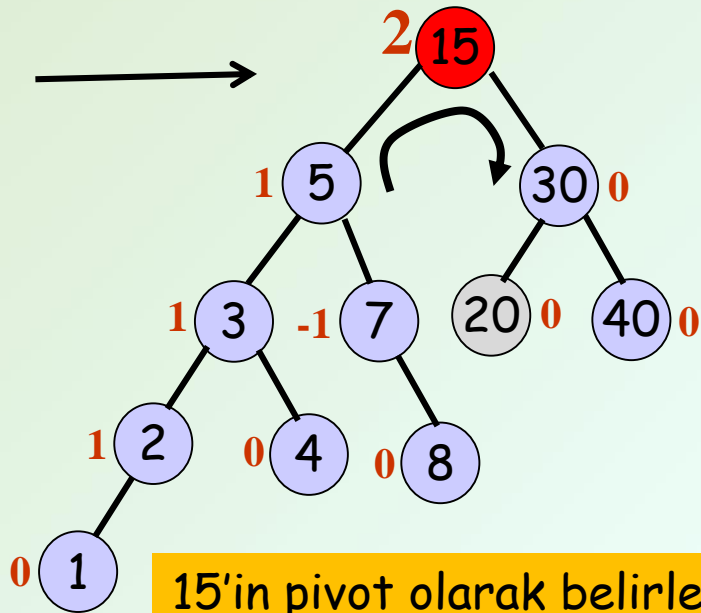
- **RR Dengesizliği:**
 - $P(20) \rightarrow bf = -2$
 - $R(30) \rightarrow bf = 0$ veya -1



Yukardaki AVL ağacı mıdır?

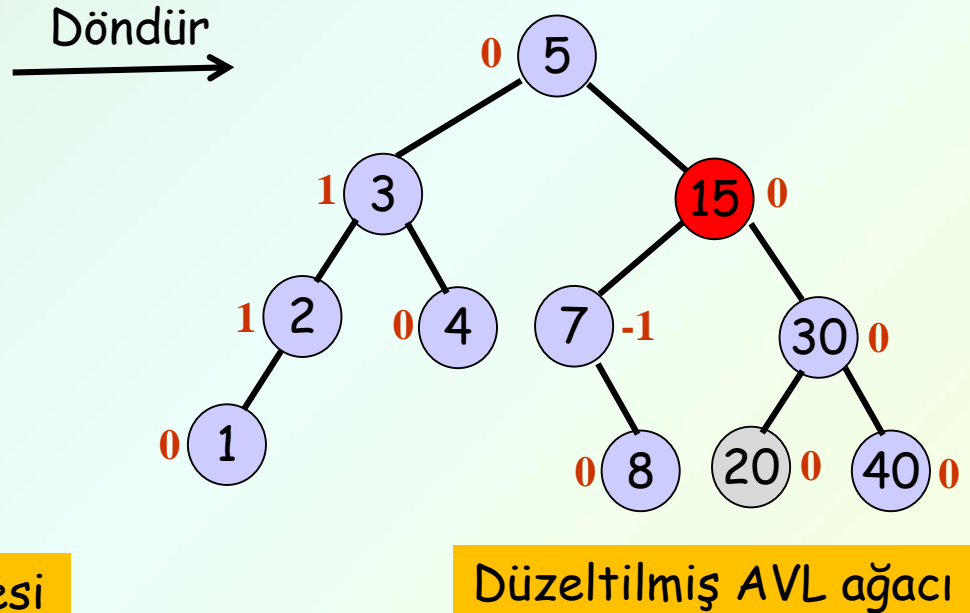
Balans faktörü düzelterek köke doğru ilerle.

Silme Örneği (3) - devam



Dengesizliğin türünü belirle

- LL Dengesizliği:
 - $P(15) \rightarrow bf = 2$
 - $L(5) \rightarrow bf = 0$ veya 1



Arama (Bul)

- AVL ağacı bir tür İkili Arama Ağacı (BST) olduğundan arama algoritması BST ile aynıdır.
- $O(\log N)$ de çalışması garantidir.

AVL Ağaçları - Özet

- AVL Ağaçlarının Avantajları
 1. AVL ağacı devamlı dengeli olduğu için Arama/Ekleme/Silme işlemi $O(\log N)$ yapılır.
 2. Düzeltme işlemi algoritmaların karmaşıklığını etkilemez.
- AVL Ağaçlarının Dezavantajları:
 1. Balans faktörü için ekstra yer gereklidir.
 2. Arı dengeli ağaçlarda AVL ağaçları kadar iyi performans verir ayrıca bu tür algoritmalarda balans faktör kullanılmaz
 - Splay ağaçları