



Bölüm 10: Arama Ağaçları

Veri Yapıları



Arama Ağacı

- Arama ağaçları, veri öğelerini hiyerarşik bir yapı içinde saklamak ve hızlı bir şekilde aramak için kullanılan bir veri yapısıdır.
- Her düğüm, belirli bir öğeyi veya anahtarı temsil eder ve bu anahtarlar bir düzen içinde yerleştirilir.
- Arama ağaçlarının ortalama arama karmaşıklığı $O(\log n)$ veya $O(h)$ olarak kabul edilir, burada "n" ağaçtaki düğüm sayısını, "h" ise ağacın yüksekliğini temsil eder.



Temel İşlemler

- **Ekleme (Insertion):** Bir öğeyi ağaca ekler.
- **Arama (Search):** Belirli bir anahtarla öğeyi arar ve döndürür.
- **Silme (Deletion):** Belirli bir anahtara sahip öğeyi ağaçtan kaldırır.
- **Dolaşma (Traversal):** Ağaç içindeki öğeleri belirli bir sırayla gezme.



Avantajlar ve Dezavantajlar

- Avantajlar:
 - Hızlı arama (Ortalama $O(\log n)$ karmaşıklık).
 - Veriyi düzenli ve sıralı bir şekilde saklar.
- Dezavantajlar:
 - Ekleme ve silme işlemleri dengelemeyi gerektirebilir.
 - Bozuk bir denge, performans sorunlarına neden olabilir.



Kullanım Alanları

- **Veritabanları:** Dizinleme ve sorgulama işlemleri.
- **Sembol tabloları:** Değişkenler ve fonksiyonlar için hızlı erişim.
- **Hafıza yönetimi:** Hafıza tahsis ve geri dönüşüm işlemleri.



Arama Ağacı Türleri

- **İkili Arama Ağaçları (Binary Search Trees):** Her düğüm, sol alt ağaçtaki düğümlerden daha küçük ve sağ alt ağaçtaki düğümlerden daha büyük olan bir anahtarı temsil eder.
- **Kırmızı-Siyah Ağaçlar (Red-Black Trees):** İkili arama ağaçlarının dengelemesi ve performansını iyileştiren bir tür arama ağacı.
- **AVL Ağaçları:** Daha sıkı bir dengeleme ile performansı optimize eden arama ağacı türü.



Arama Ağacı Türleri

- **İkili Arama Ağaçları (Binary Search Trees):** Her düğüm, sol alt ağaçtaki düğümlerden daha küçük ve sağ alt ağaçtaki düğümlerden daha büyük olan bir anahtarı temsil eder.
- **Kırmızı-Siyah Ağaçlar (Red-Black Trees):** İkili arama ağaçlarının dengelemesi ve performansını iyileştiren bir tür arama ağacı.
- **AVL Ağaçları:** Daha sıkı bir dengeleme ile performansı optimize eden arama ağacı türü.



İkili Arama Ağacı

- İkili arama ağacı, her düğümün en fazla iki alt düğümü olan bir ağaç yapısıdır.
- Her düğüm, sol alt ağaçtaki düğümlerden daha küçük ve sağ alt ağaçtaki düğümlerden daha büyük olan bir anahtarı temsil eder.
- Bu özelliği sayesinde hızlı arama işlemleri sağlar.



Özellikler

- İkili arama ağacının temel özellikleri şunlardır:
 - Her düğümün en fazla iki çocuğu vardır.
 - Sol çocuk, düğümün anahtarından daha küçük değerlere sahiptir.
 - Sağ çocuk, düğümün anahtarından daha büyük değerlere sahiptir.



Arama Ağacı Türleri

- **İkili Arama Ağaçları (Binary Search Trees):** Her düğüm, sol alt ağaçtaki düğümlerden daha küçük ve sağ alt ağaçtaki düğümlerden daha büyük olan bir anahtarı temsil eder.
- **Kırmızı-Siyah Ağaçlar (Red-Black Trees):** İkili arama ağaçlarının dengelemesi ve performansını iyileştiren bir tür arama ağacı.
- **AVL Ağaçları:** Daha sıkı bir dengeleme ile performansı optimize eden arama ağacı türü.



Kırmızı-Siyah Ağaçlar

- Kırmızı-Siyah ağaçları, veri yapıları içinde dengeli bir ağaç yapısı sağlayan önemli bir veri yapısıdır.
- Kırmızı-Siyah ağaçları, her düğümün üzerine kırmızı veya siyah bir renk etiketi taşıdığı bir tür ikili arama ağacıdır.
- Ağaç, özel kurallara tabidir ve bu kurallar ağacın dengeli kalmasını sağlar.



Temel Özellikler

- Her düğüm kırmızı veya siyah renkte olabilir.
- Kök düğüm siyah renkte olmalıdır.
- Her yaprak düğümün (NIL düğümleri) siyah olması gerekmektedir.
- Ardışık kırmızı düğümler kabul edilmez.



Avantajlar ve Dezavantajlar

- Avantajlar:
 - Dengeli bir ağaç yapısıdır, arama işlemleri hızlıdır (Ortalama $O(\log n)$ karmaşıklık).
 - Ekleme ve silme işlemleri dengeleme ile yapılır.
- Dezavantajlar:
 - Ekstra bellek tüketebilir ve kod karmaşıklığı yüksektir.
 - Diğer ağaç türlerine göre daha yavaştır.



Arama Ağacı Türleri

- **İkili Arama Ağaçları (Binary Search Trees):** Her düğüm, sol alt ağaçtaki düğümlerden daha küçük ve sağ alt ağaçtaki düğümlerden daha büyük olan bir anahtarı temsil eder.
- **Kırmızı-Siyah Ağaçlar (Red-Black Trees):** İkili arama ağaçlarının dengelemesi ve performansını iyileştiren bir tür arama ağacı.
- **AVL Ağaçları:** Daha sıkı bir dengeleme ile performansı optimize eden arama ağacı türü.



AVL Ağacı

- AVL ağacı, her düğümün üzerine bir denge faktörü eklenmiş bir ikili arama ağacıdır.
- Ağaç, özel bir dengeleme kuralına tabidir ve bu kural ağacın dengeli kalmasını sağlar.



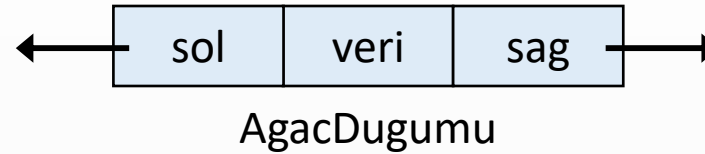
Temel Özellikler

- Her düğümün denge faktörü, sol alt ağacın yüksekliği ile sağ alt ağacın yüksekliği arasındaki farkı temsil eder.
- Her düğümün denge faktörü -1, 0 veya 1 olmalıdır.
- Ağaç dengeli olduğunda, her düğümün alt ağaçları da dengelidir.





İkili Arama Ağacında Ağaç Düğümü Yapısı



```
public class AgacDugumu {  
    private int veri;  
    private AgacDugumu sol;  
    private AgacDugumu sag;  
  
    public AgacDugumu(int veri) {  
        this.veri = veri;  
        this.sol = null;  
        this.sag = null;  
    }  
}
```

İkili Arama Ağacına Bir Değer Eklenmesi





metot çağırımı

satır no

kok

sonuc

sol

sag

```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10         kok.sag = ekle(kok.sag, deger);
11     return kok;
12 }
```



metot çağırımı

satır no

kok

sonuc

sol

sag

kok → null

```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10         kok.sag = ekle(kok.sag, deger);
11     return kok;
12 }
```



kok → null



metot çağırımı	satır no	kok	sol	sag	deger
ekle		null	null	null	5

deger = 5



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```

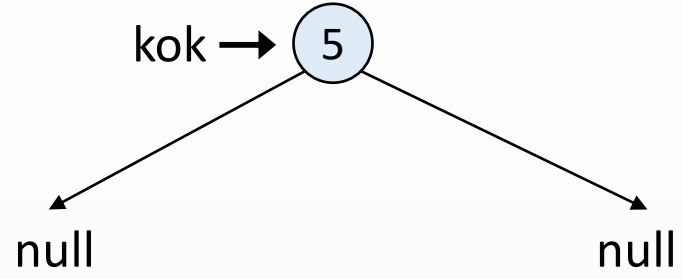


kok → null

metot çağırımı	satır no	kok	sol	sag	deger
→ ekle		null	null	null	5

deger = 5

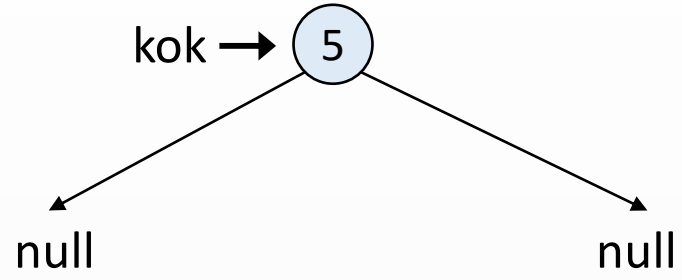
```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



deger = 5

metot çağırımı	satır no	kok	sol	sag	deger
→ ekle		5	null	null	5

```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```

deger = 5

metot çağırımı	satır no	kok	sol	sag	deger
→ ekle		5	null	null	5

```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



metot çağırımı

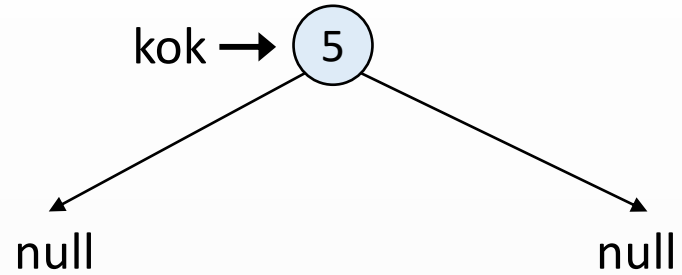
satır no

kok

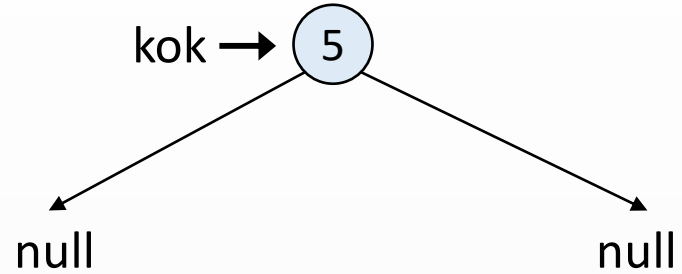
sol

sag

deger



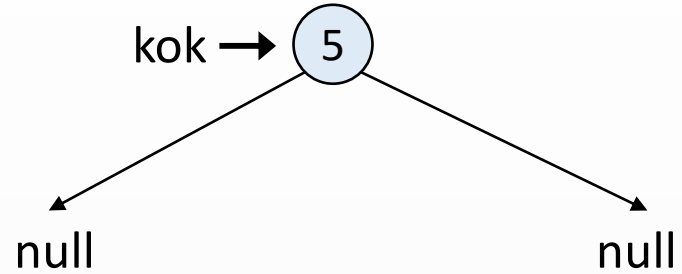
```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



deger = 3

metot çağırımı	satır no	kok	sol	sag	deger
→ ekle		5	null	null	3

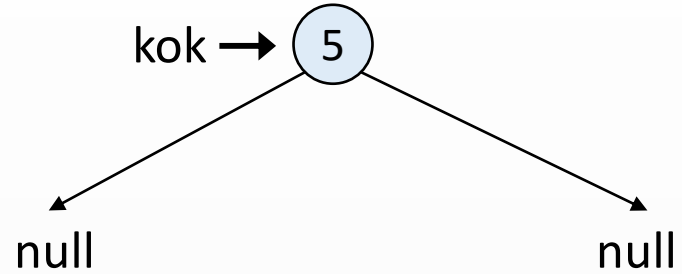
```
→ 1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



deger = 3

metot çağırımı	satır no	kok	sol	sag	deger
→ ekle		5	null	null	3

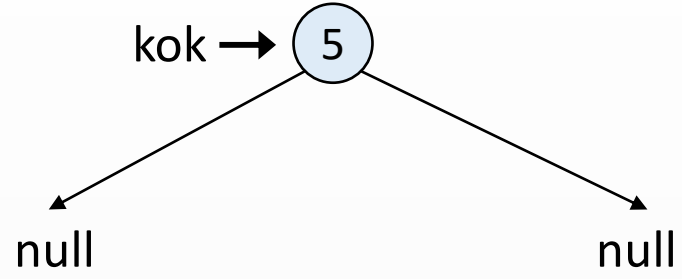
```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



deger = 3

metot çağırımı	satır no	kok	sol	sag	deger
→ ekle		5	null	null	3

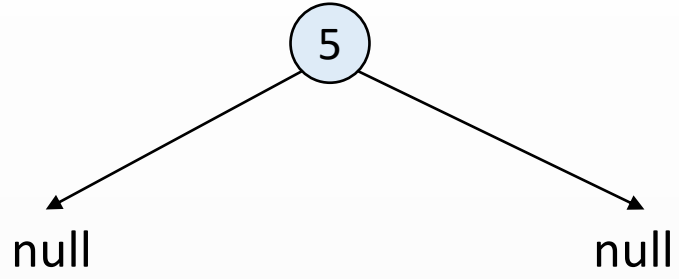
```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



deger = 3

metot çağırımı	satır no	kok	sol	sag	deger
→ ekle	8	5	null	null	3

```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



kok → null

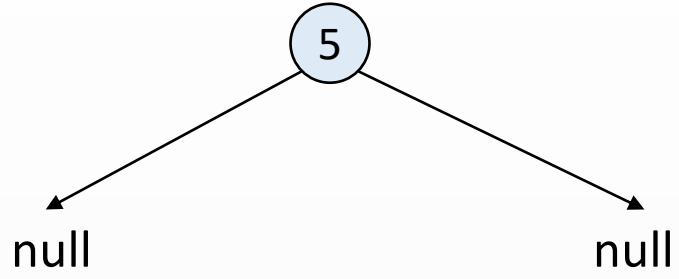
deger = 3



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	null	null	3
ekle		null	null	null	3



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



kok → null

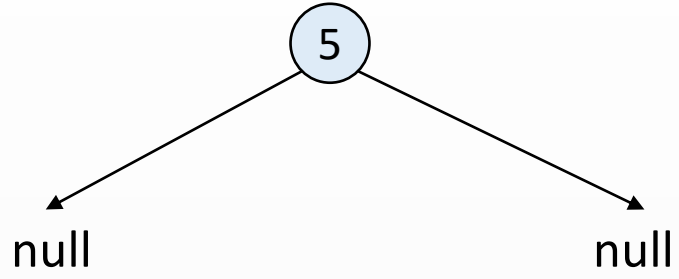
deger = 3



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	null	null	3
ekle		null	null	null	3



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```

kok → null

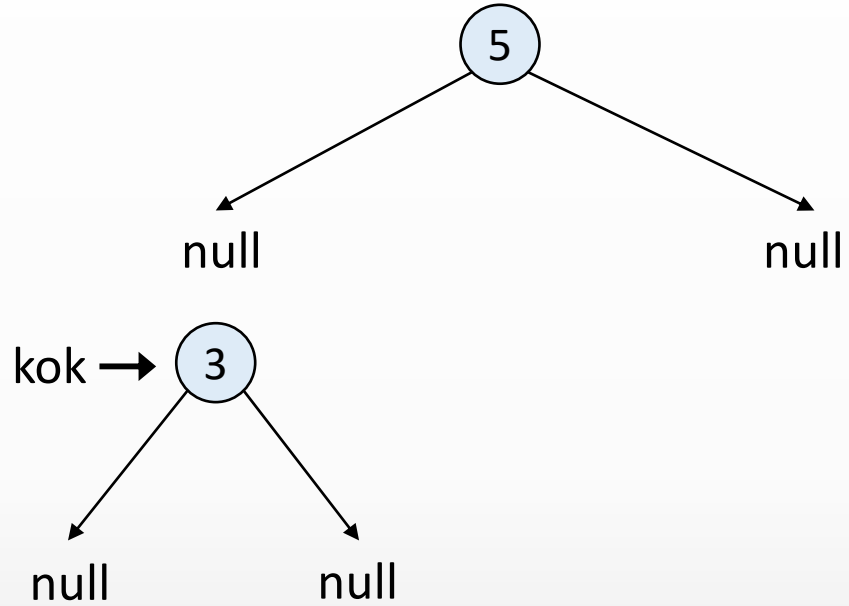
deger = 3



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	null	null	3
ekle		null	null	null	3



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



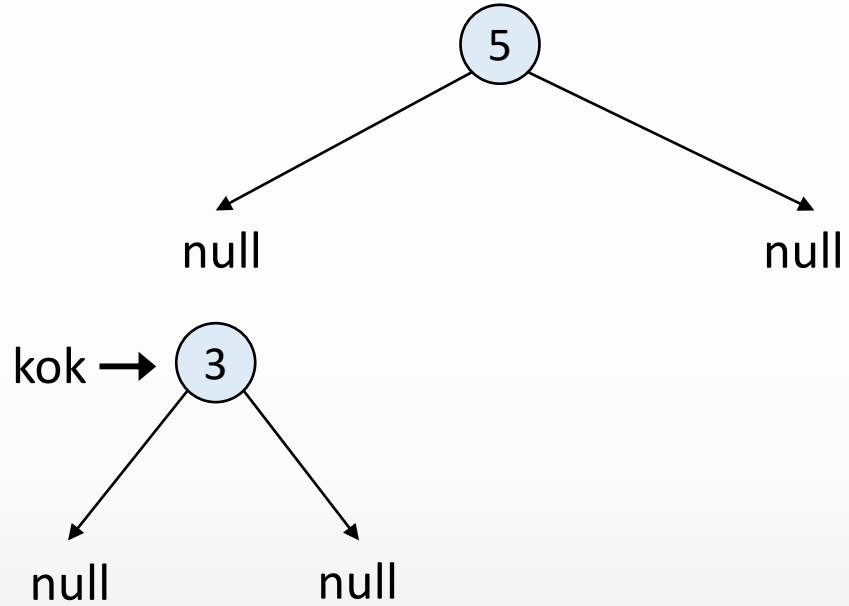
deger = 3



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	null	null	3
ekle		3	null	null	3



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



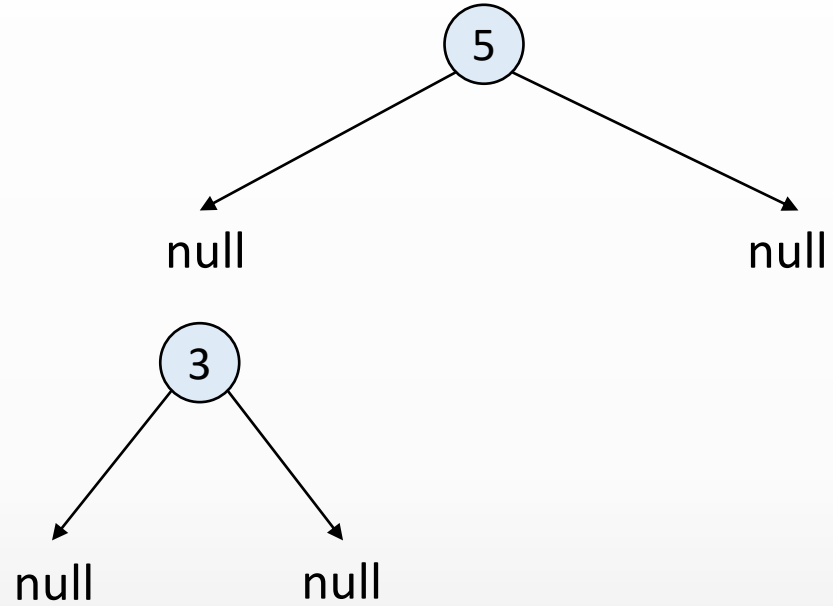
deger = 3



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	null	null	3
ekle		3	null	null	3

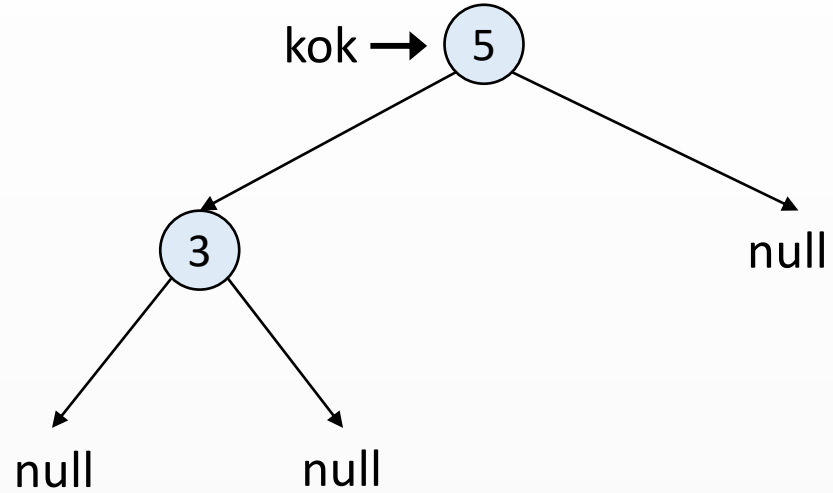


```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	null	null	3

```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



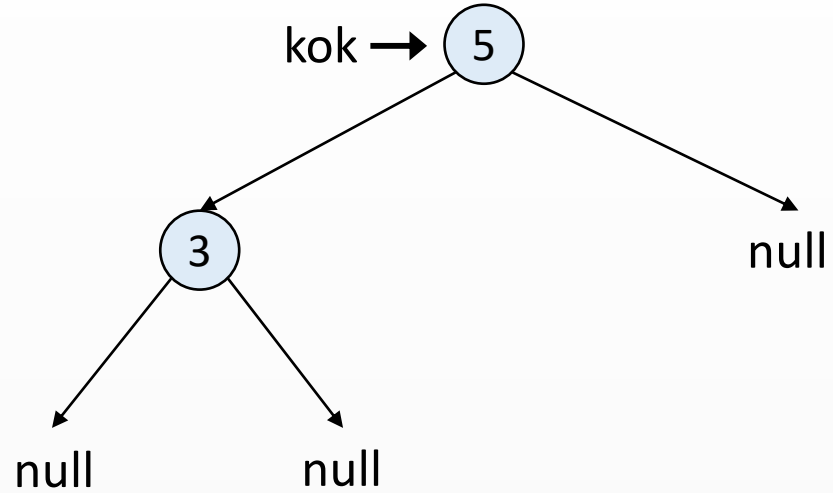
deger = 3



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	3	null	3



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	3	null	3



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



metot çağırımı

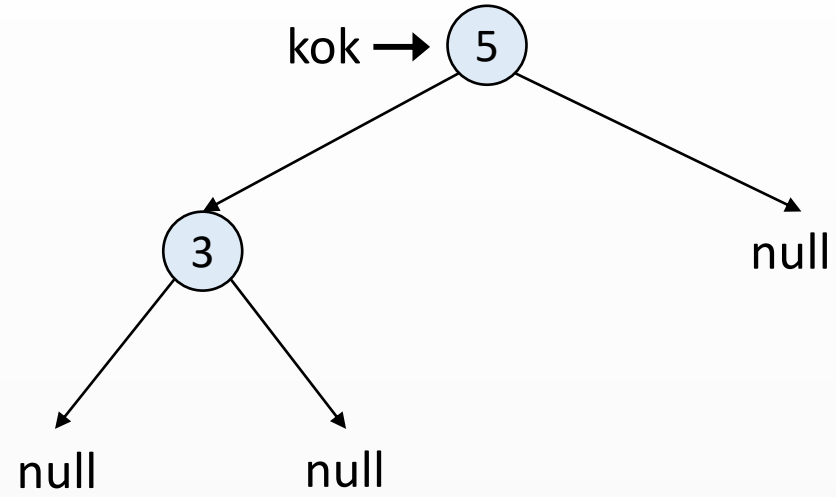
satır no

kok

sol

sag

deger



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



metot çağırımı

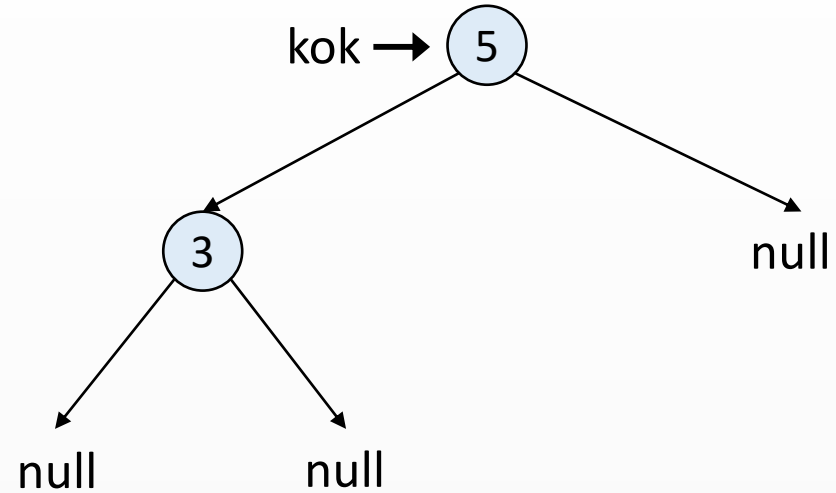
satır no

kok

sol

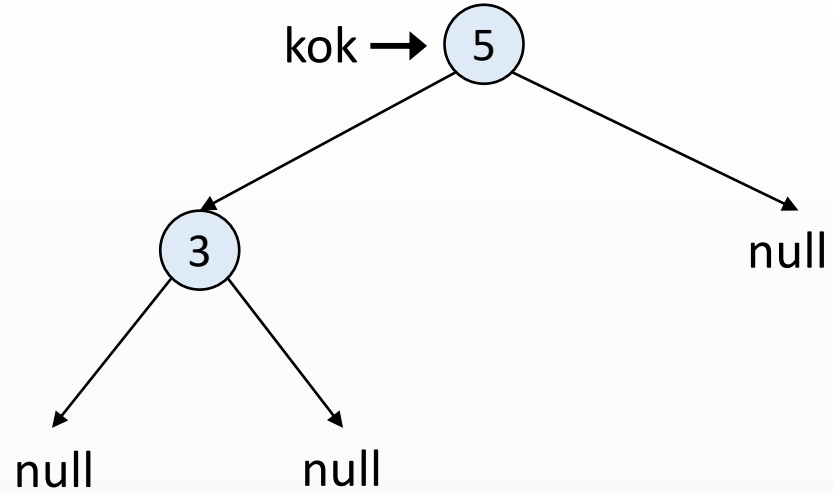
sag

deger



deger = 7

```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```

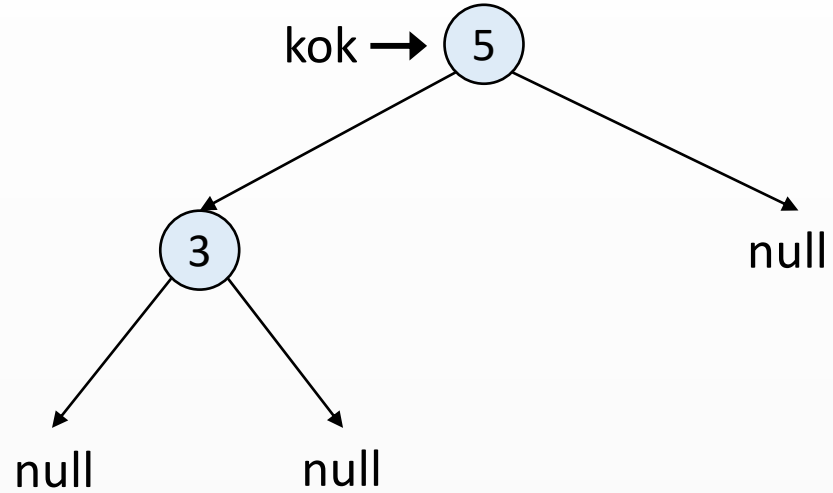
deger = 7



metot çağırımı	satır no	kok	sol	sag	deger
ekle		5	3	null	7



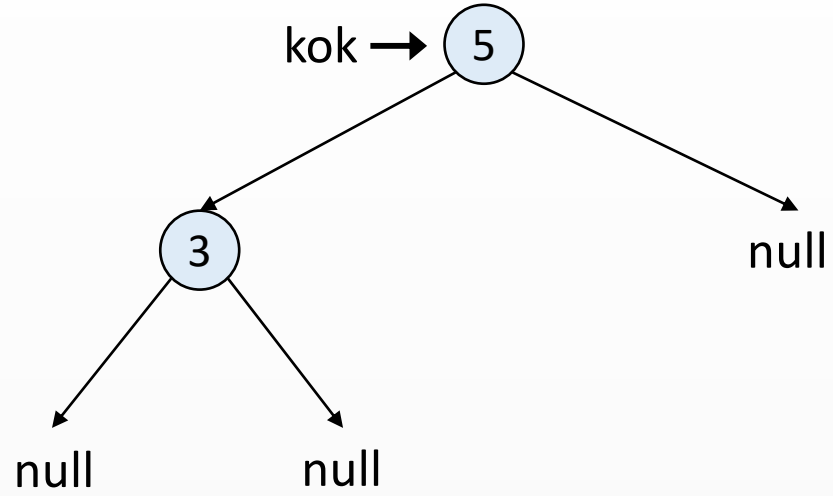
```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



deger = 7

metot çağırımı	satır no	kok	sol	sag	deger
→ ekle		5	3	null	7

```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



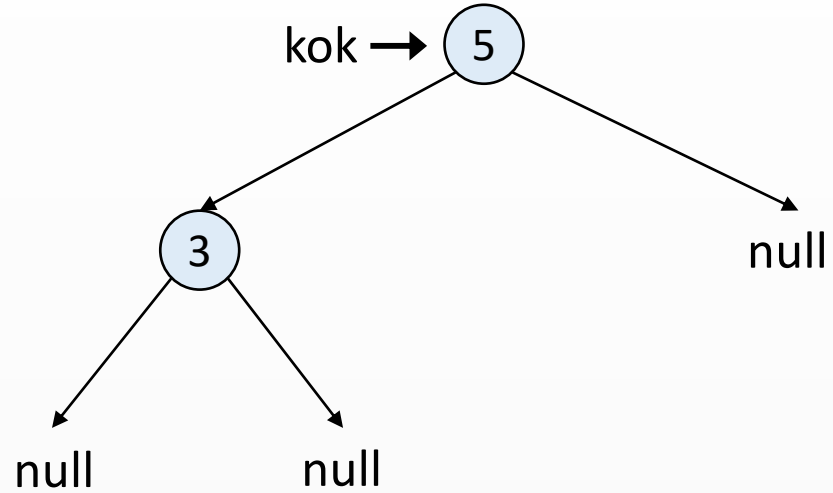
deger = 7



metot çağırımı	satır no	kok	sol	sag	deger
ekle		5	3	null	7



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



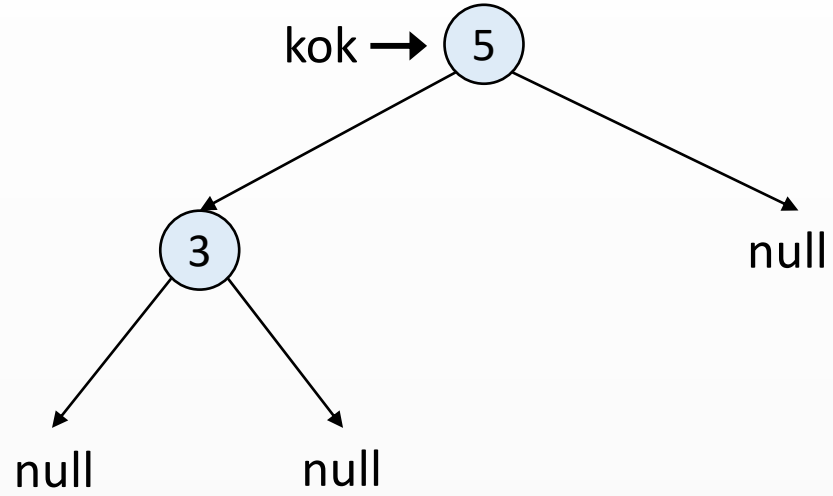
deger = 7



metot çağırımı	satır no	kok	sol	sag	deger
ekle		5	3	null	7



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



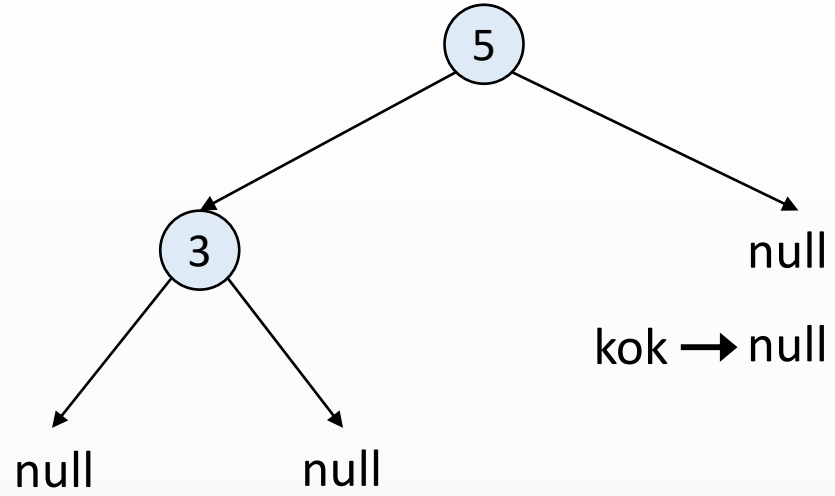
deger = 7



metot çağırımı	satır no	kok	sol	sag	deger
ekle	10	5	3	null	7



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



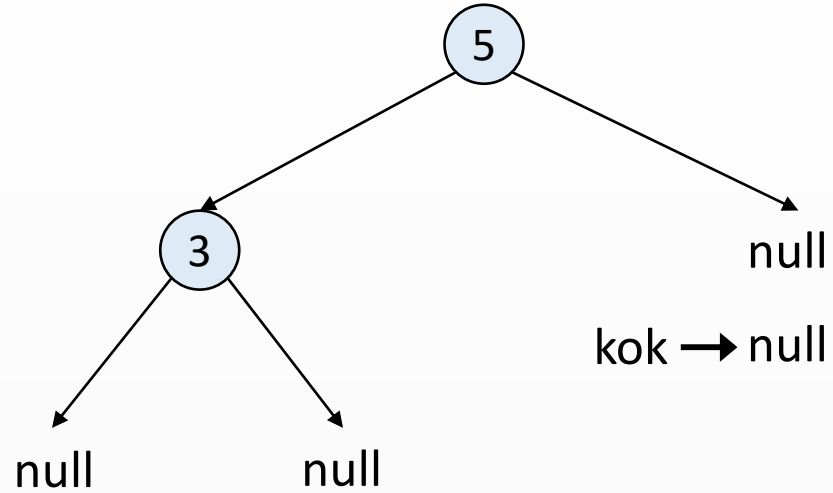
deger = 7



metot çağırımı	satır no	kok	sol	sag	deger
ekle	10	5	3	null	7
ekle		null	null	null	7



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



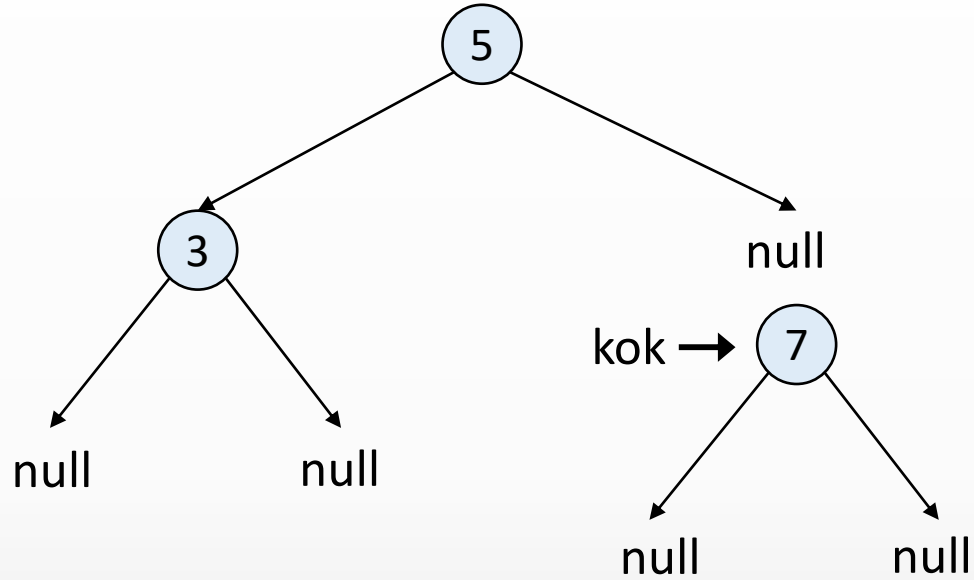
deger = 7



metot çağırımı	satır no	kok	sol	sag	deger
ekle	10	5	3	null	7
ekle		null	null	null	7



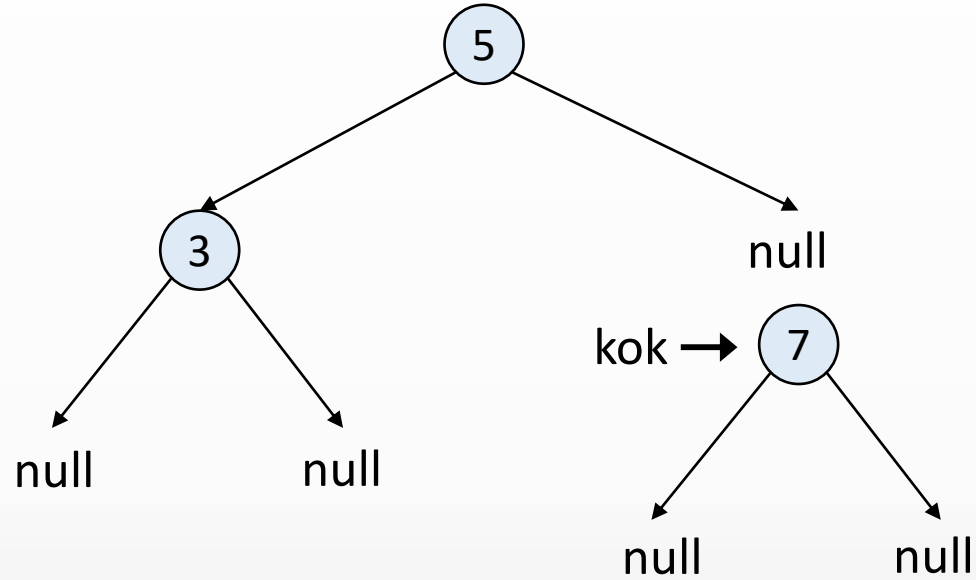
```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



deger = 7

metot çağırımı	satır no	kok	sol	sag	deger
ekle	10	5	3	null	7
ekle		7	null	null	7

```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```

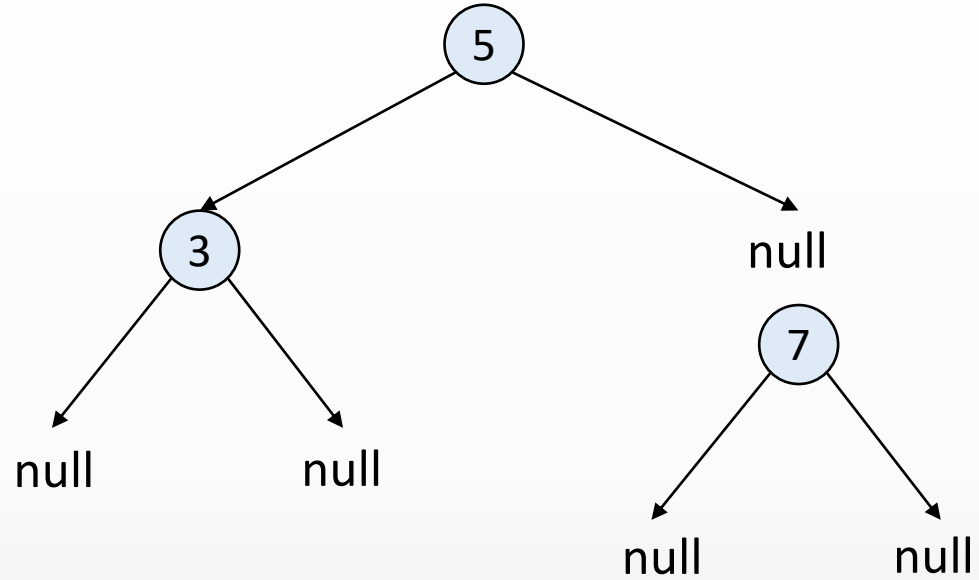
deger = 7



metot çağırımı	satır no	kok	sol	sag	deger
ekle	10	5	3	null	7
ekle		7	null	null	7



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```

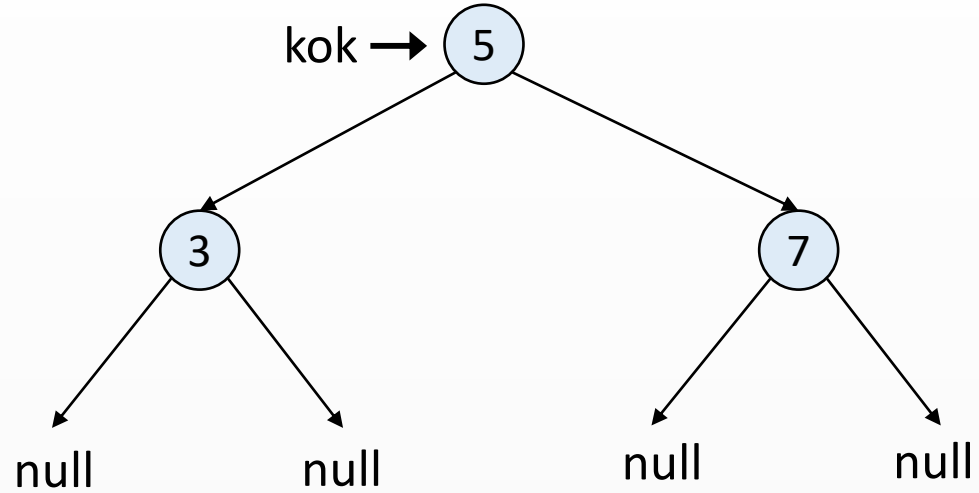


deger = 7



metot çağırımı	satır no	kok	sol	sag	deger
ekle	10	5	3	null	7

```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



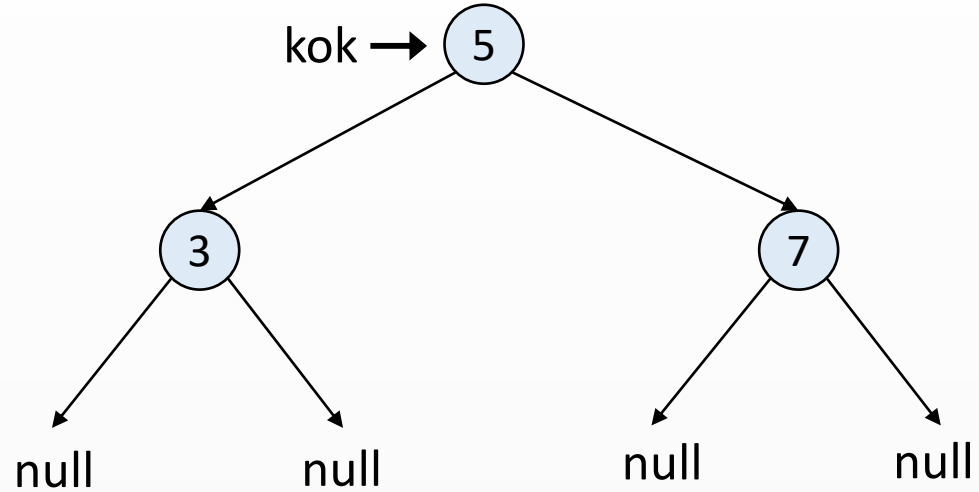
deger = 7



metot çağırımı	satır no	kok	sol	sag	deger
ekle	10	5	3	7	7



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



deger = 7



metot çağırımı	satır no	kok	sol	sag	deger
ekle	10	5	3	7	7



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



metot çağırımı

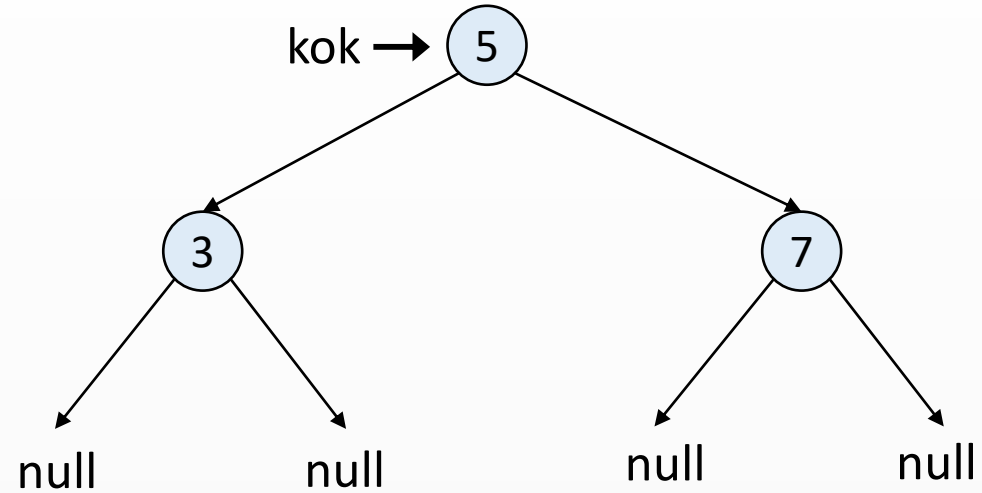
satır no

kok

sol

sag

deger



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



metot çağırımı

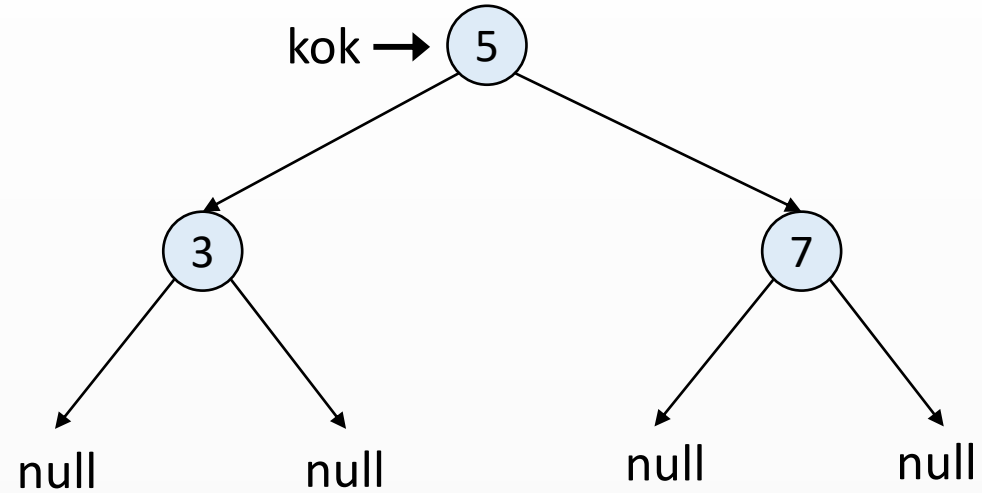
satır no

kok

sol

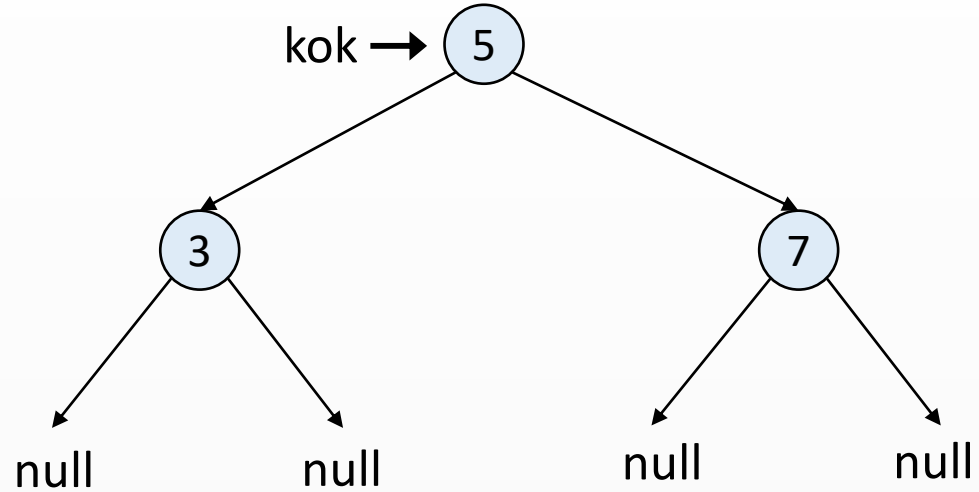
sag

deger



deger = 1

```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



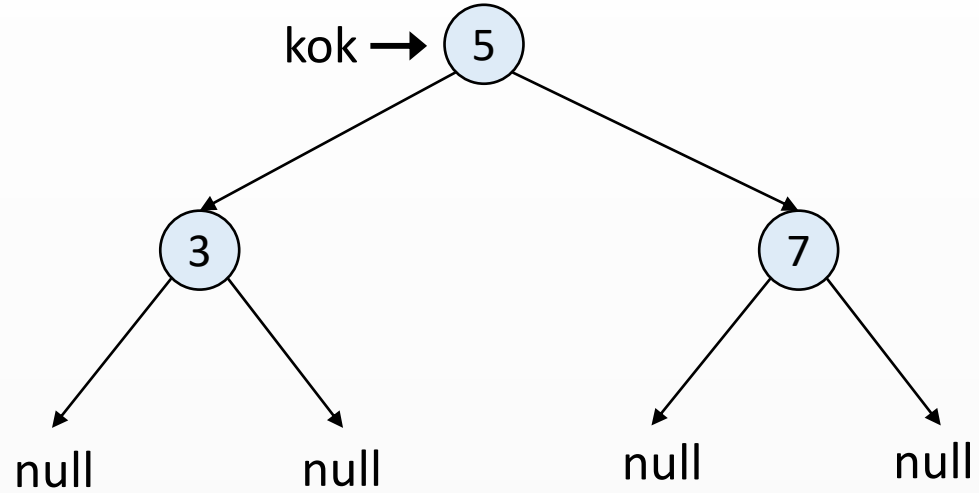
deger = 1



metot çağırımı	satır no	kok	sol	sag	deger
ekle		5	3	7	1



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



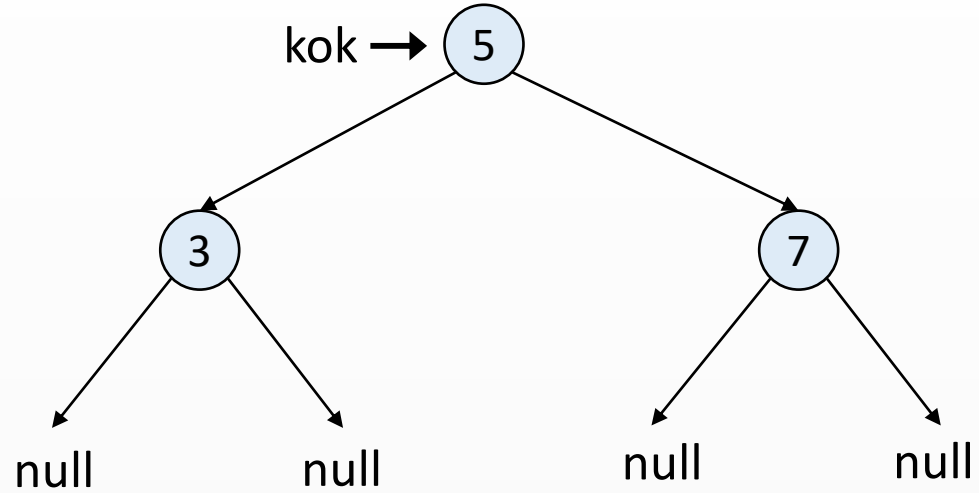
deger = 1



metot çağırımı	satır no	kok	sol	sag	deger
ekle		5	3	7	1



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```

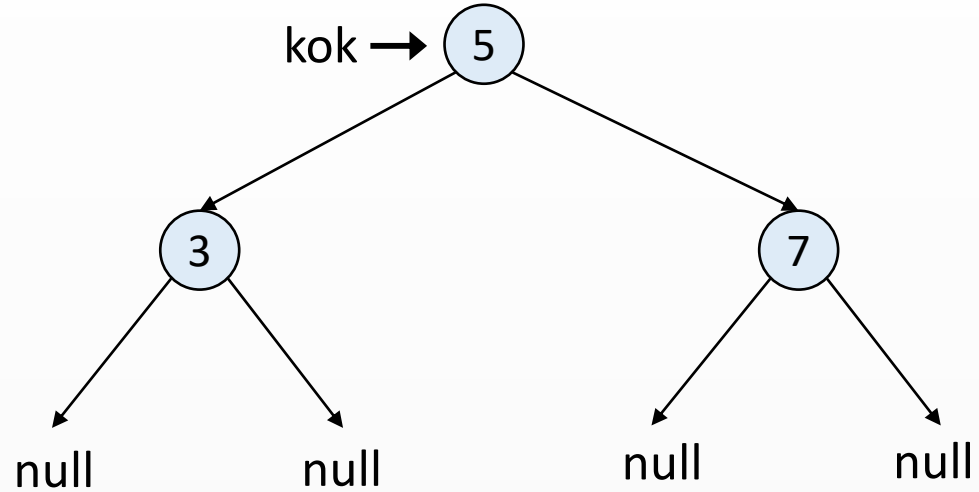
deger = 1



metot çağırımı	satır no	kok	sol	sag	deger
ekle		5	3	7	1



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



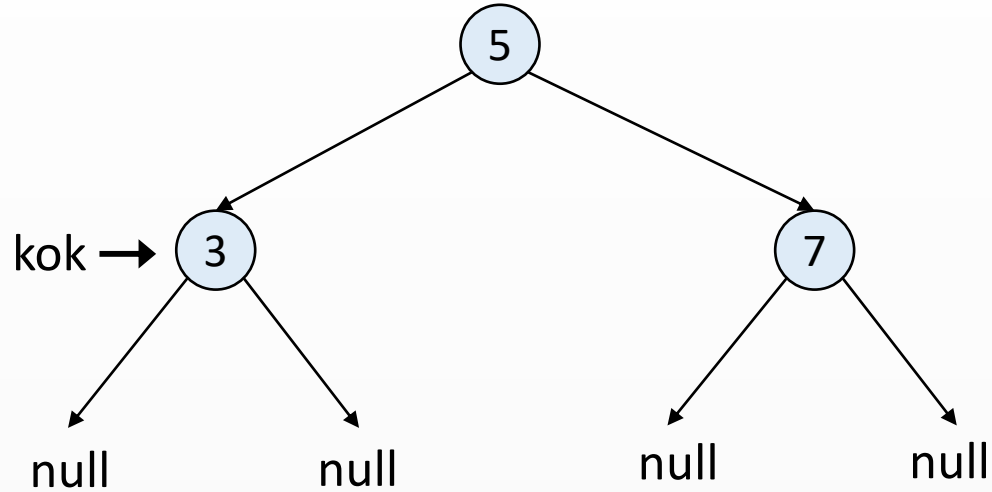
deger = 1



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	3	7	1



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



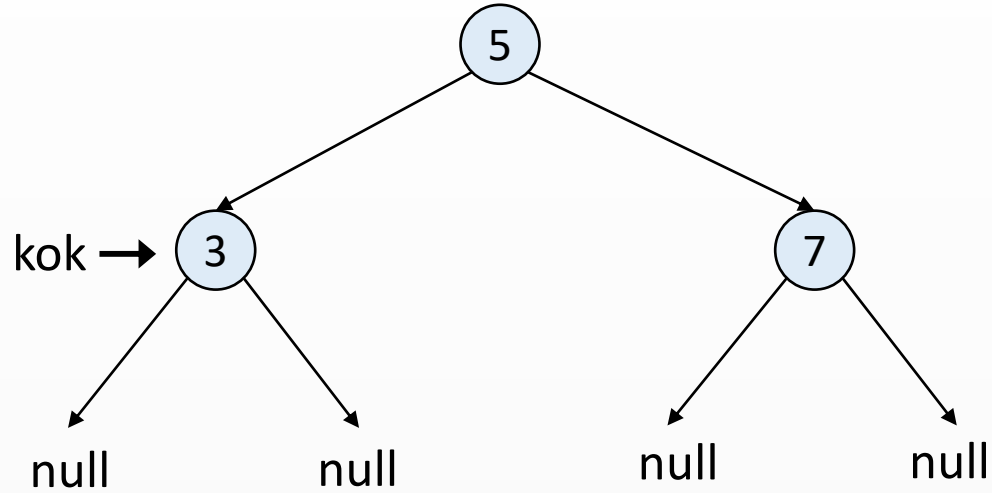
deger = 1



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	3	7	1
ekle		3	null	null	1



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



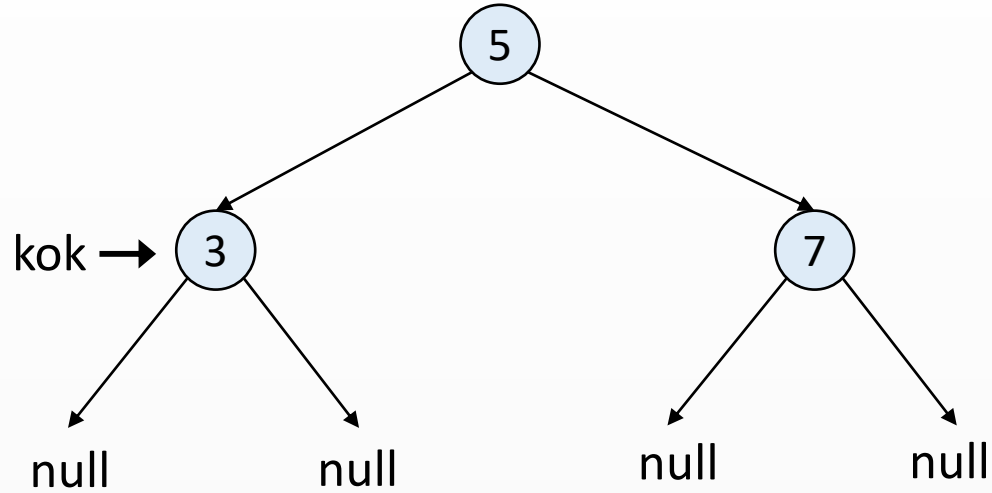
deger = 1



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	3	7	1
ekle		3	null	null	1



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



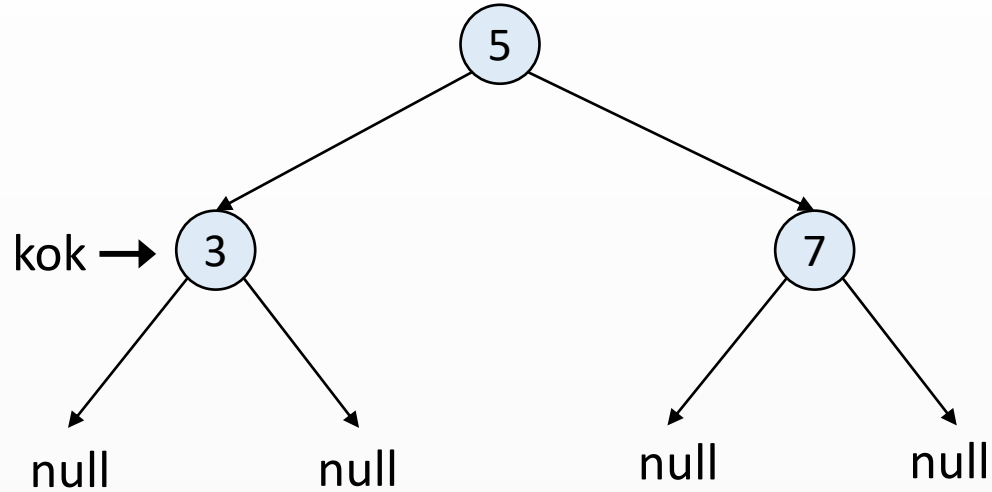
deger = 1



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	3	7	1
ekle		3	null	null	1



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



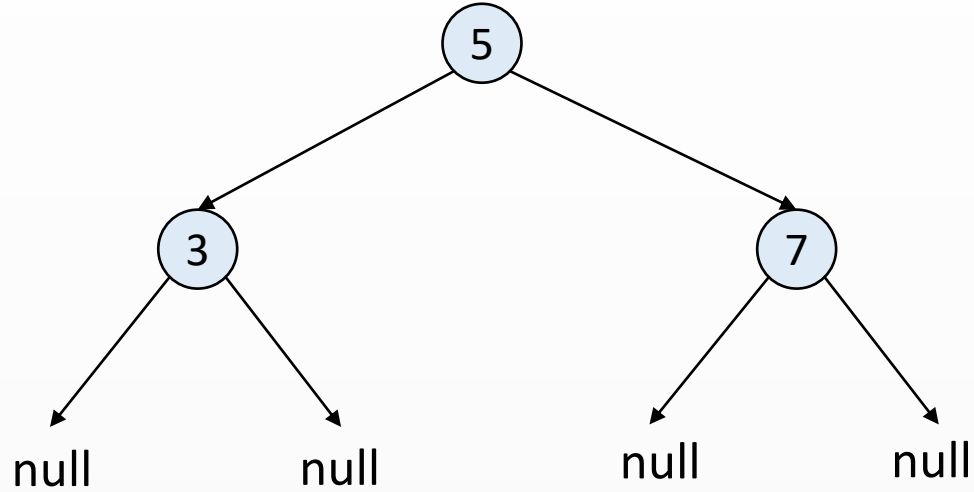
deger = 1



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	3	7	1
ekle	8	3	null	null	1



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



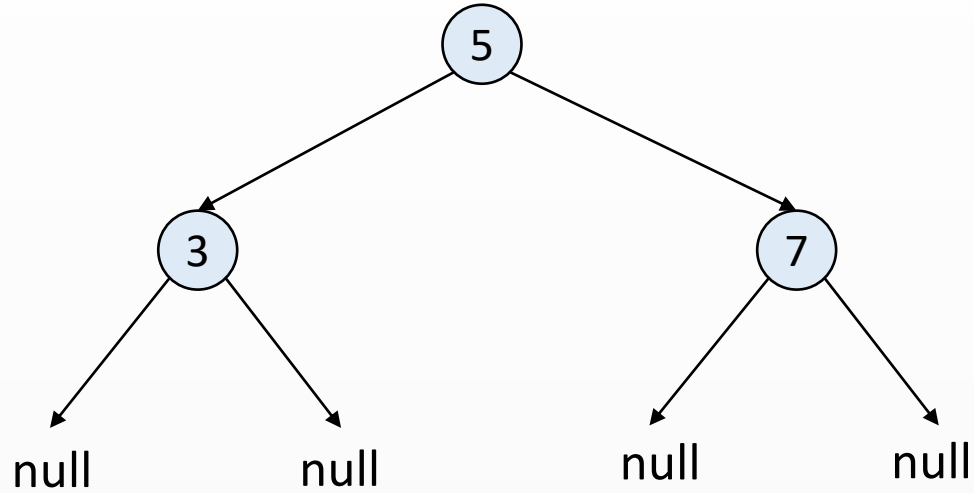
metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	3	7	1
ekle	8	3	null	null	1
ekle		null	null	null	1



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```

kok → null

deger = 1



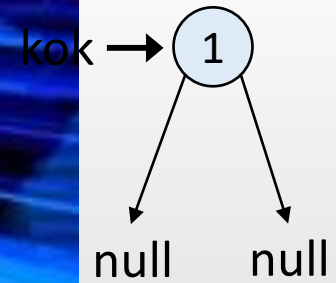
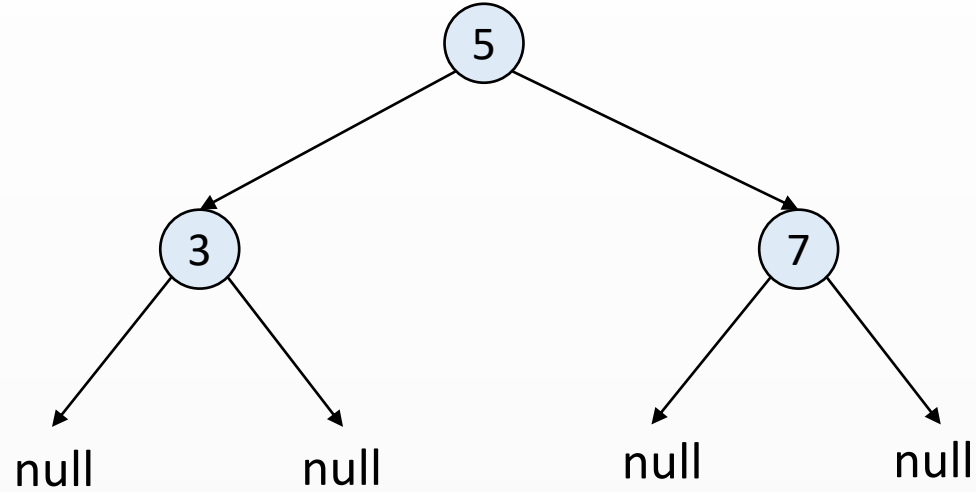
metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	3	7	1
ekle	8	3	null	null	1
ekle		null	null	null	1



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```

kok → null

deger = 1



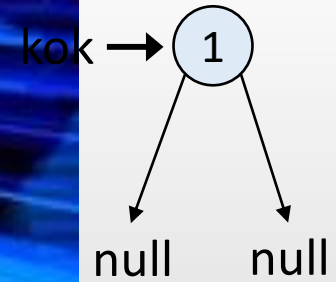
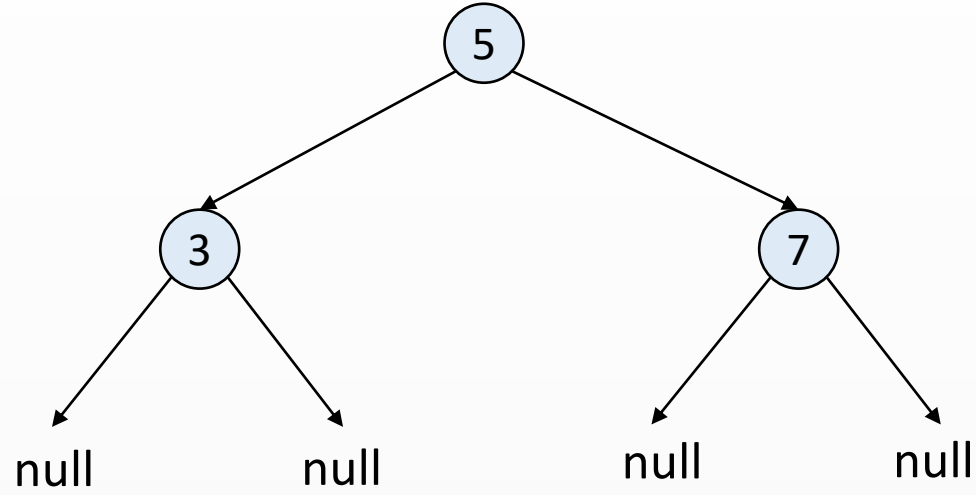
deger = 1



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	3	7	1
ekle	8	3	null	null	1
ekle		null	null	null	1



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



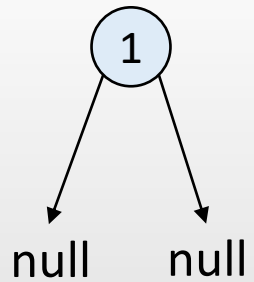
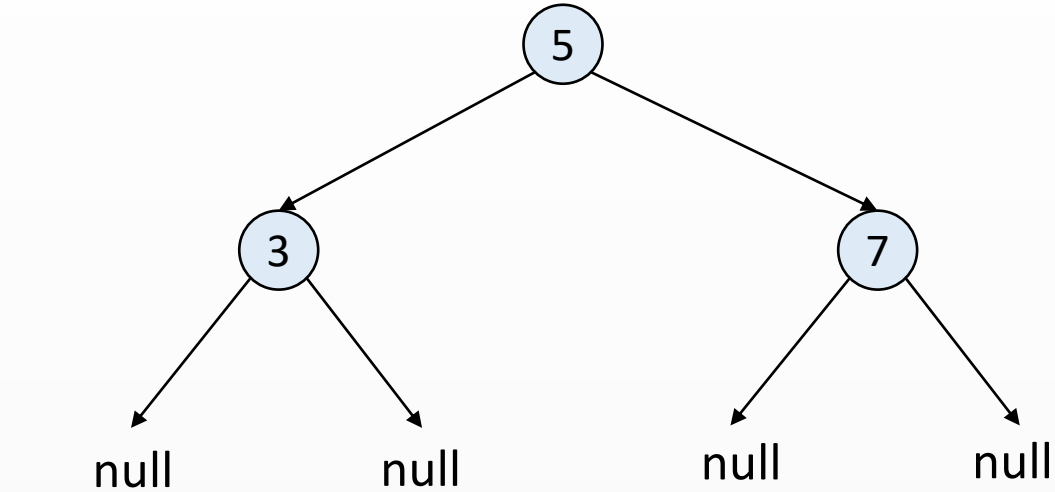
deger = 1



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	3	7	1
ekle	8	3	null	null	1
ekle		null	null	null	1



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```

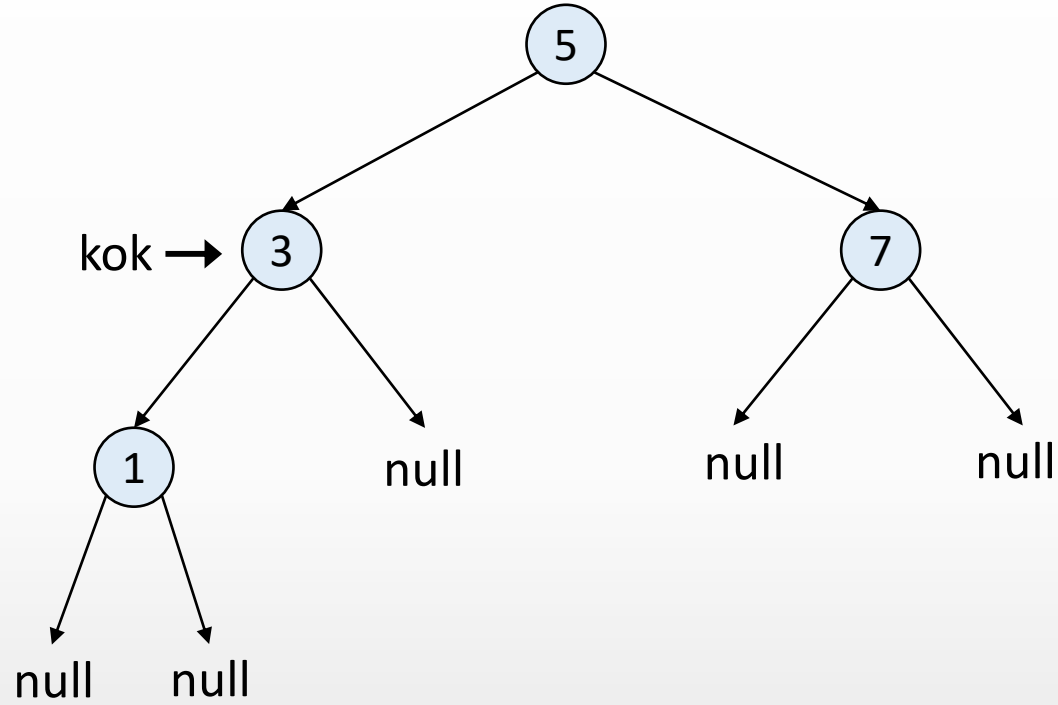


deger = 1



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	3	7	1
ekle	8	3	null	null	1

```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



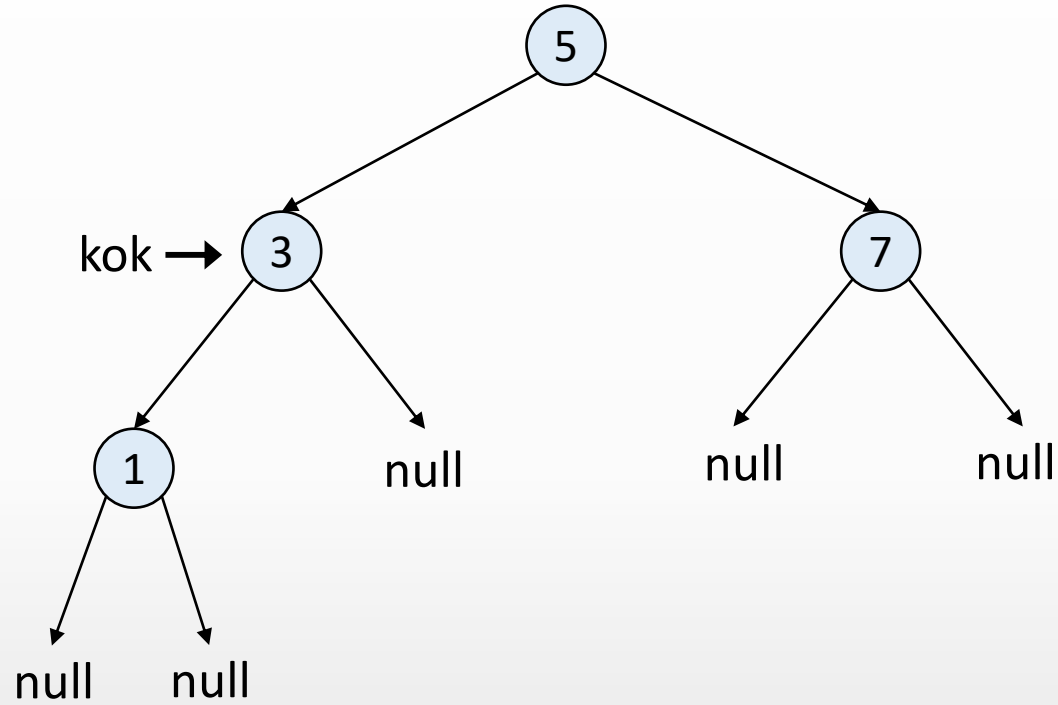
deger = 1



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	3	7	1
ekle	8	3	1	null	1



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



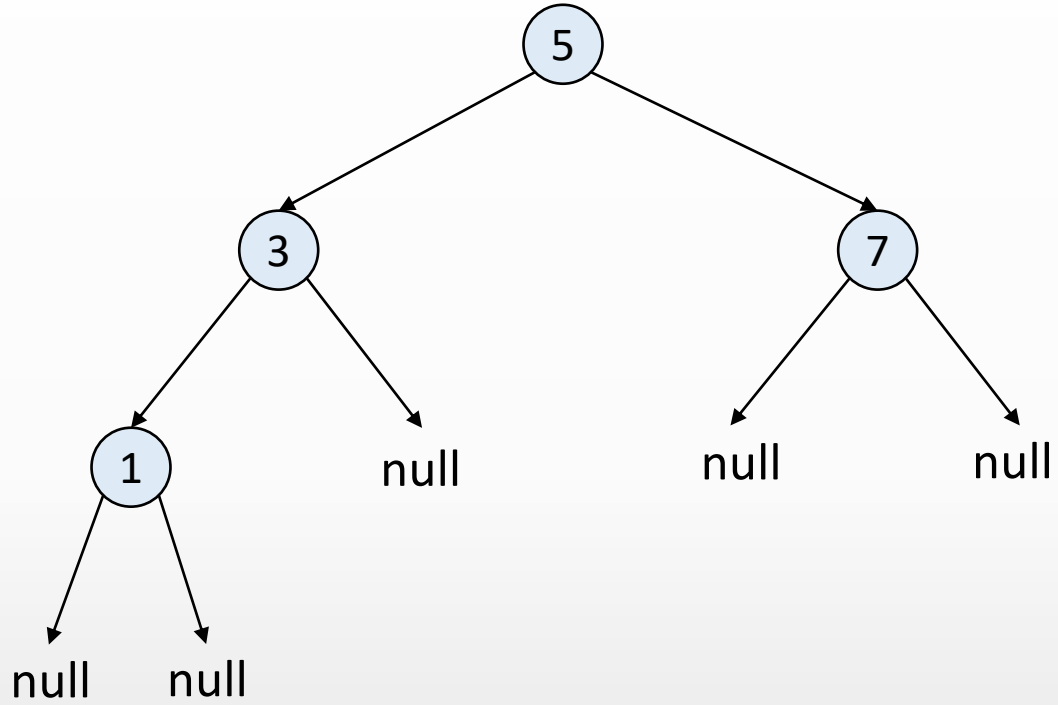
deger = 1



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	3	7	1
ekle	8	3	1	null	1



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```

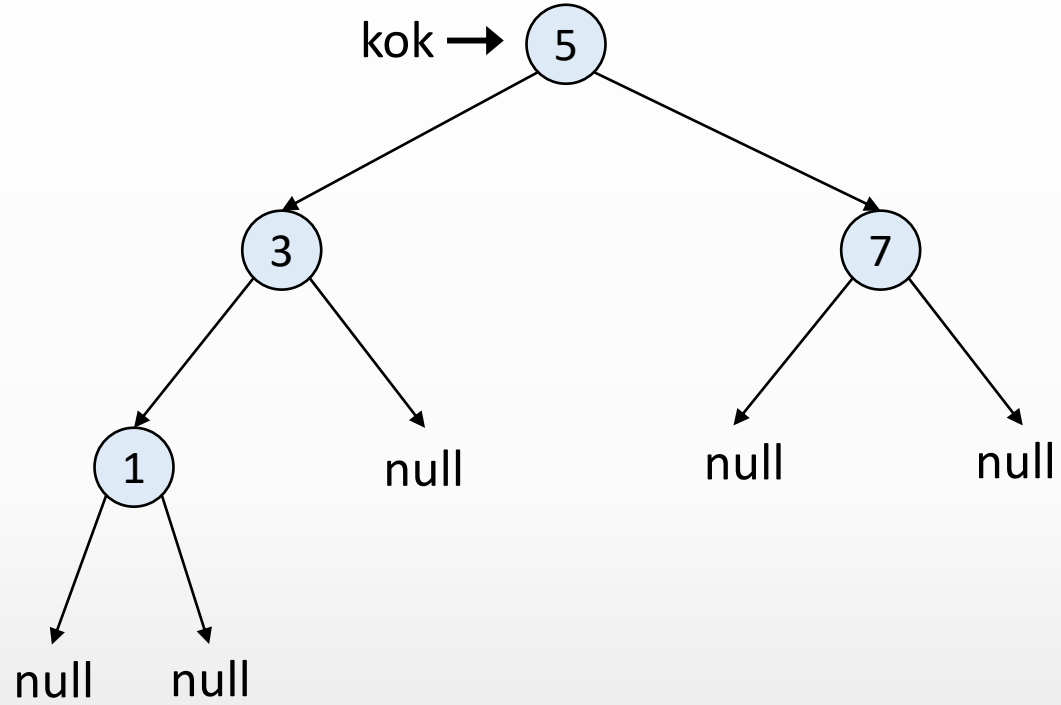


deger = 1



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	3	7	1

```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



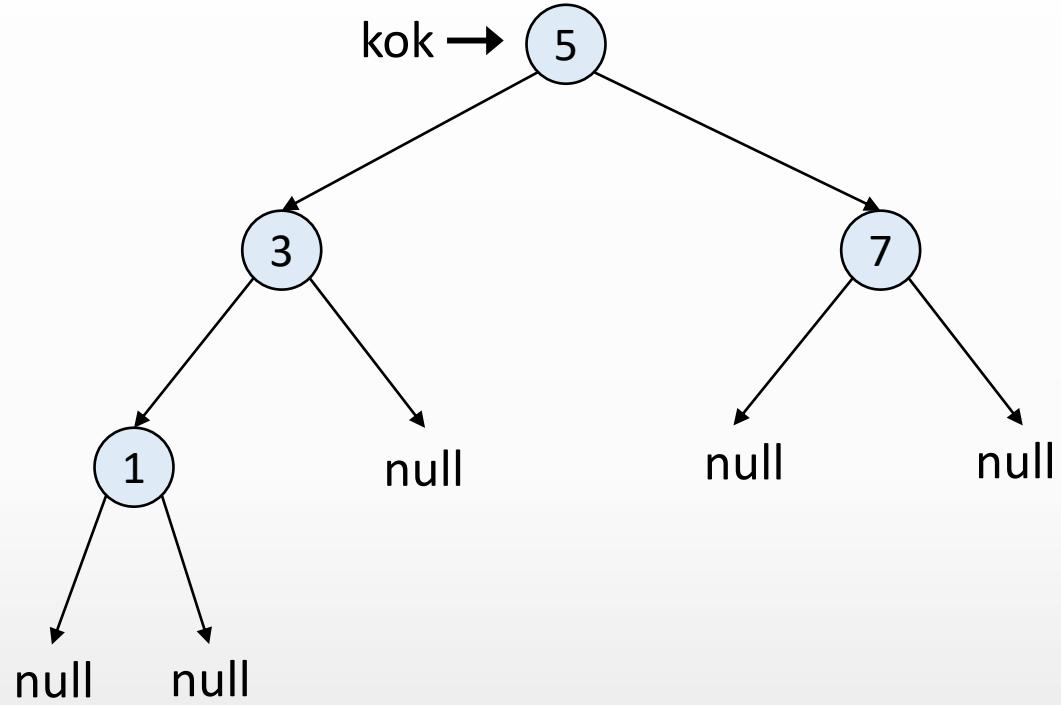
deger = 1



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	3	7	1



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```



deger = 1



metot çağırımı	satır no	kok	sol	sag	deger
ekle	8	5	3	7	1



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```




metot çağırımı

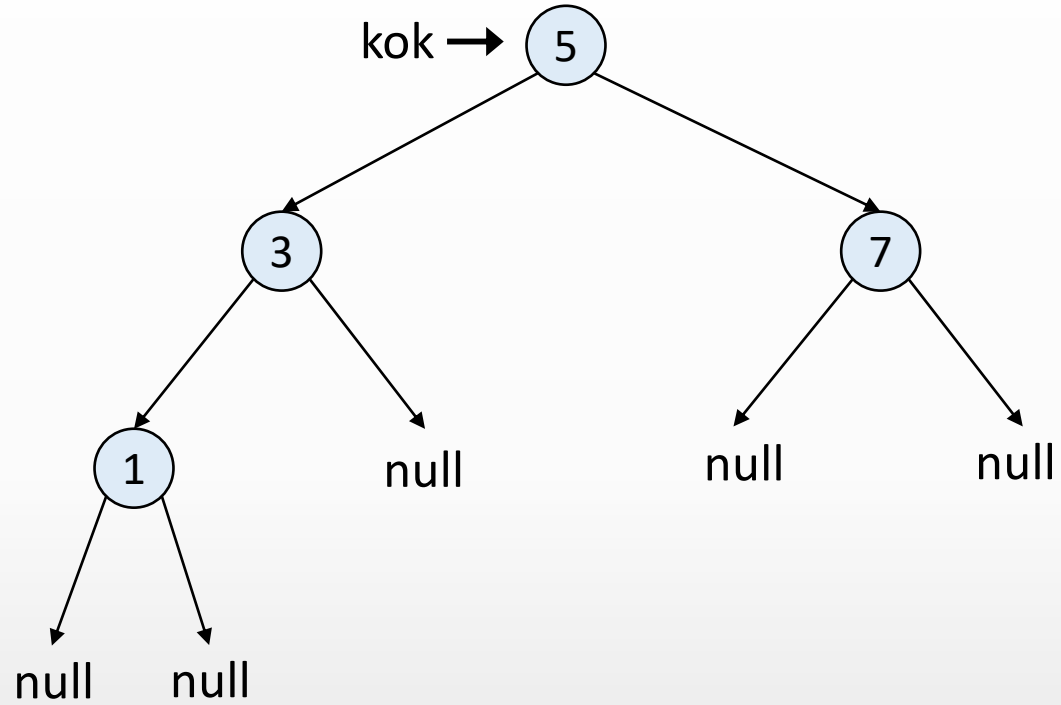
satır no

kok

sol

sag

deger



```
1 public AgacDugumu ekle(AgacDugumu kok, int deger)
2 {
3     if(kok == null) {
4         kok = new AgacDugumu(deger);
5         return kok;
6     }
7     if(deger < kok.veri)
8         kok.sol = ekle(kok.sol, deger);
9     else
10        kok.sag = ekle(kok.sag, deger);
11    return kok;
12 }
```

Verilen Bir Anahtarı İkili Arama Ağacında Arama



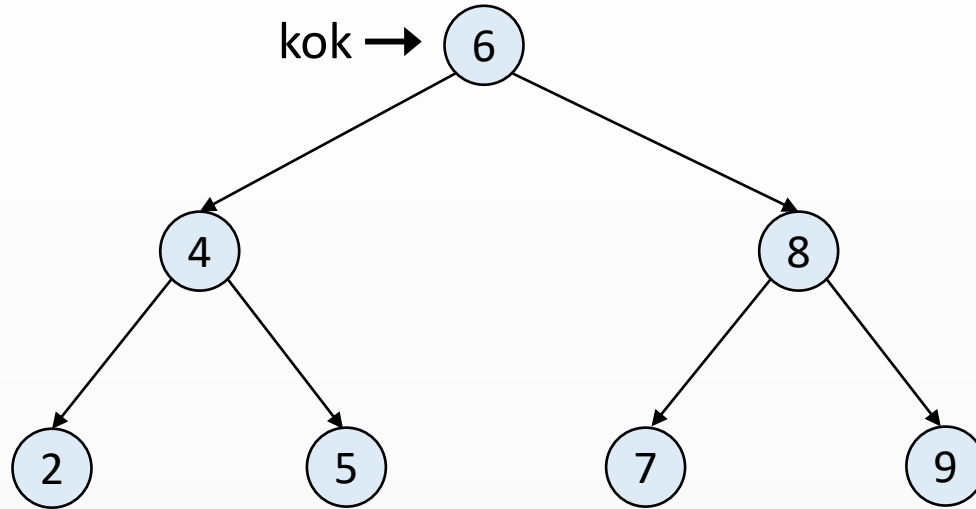


metot çağırımı

satır no

kok

anahtar



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```

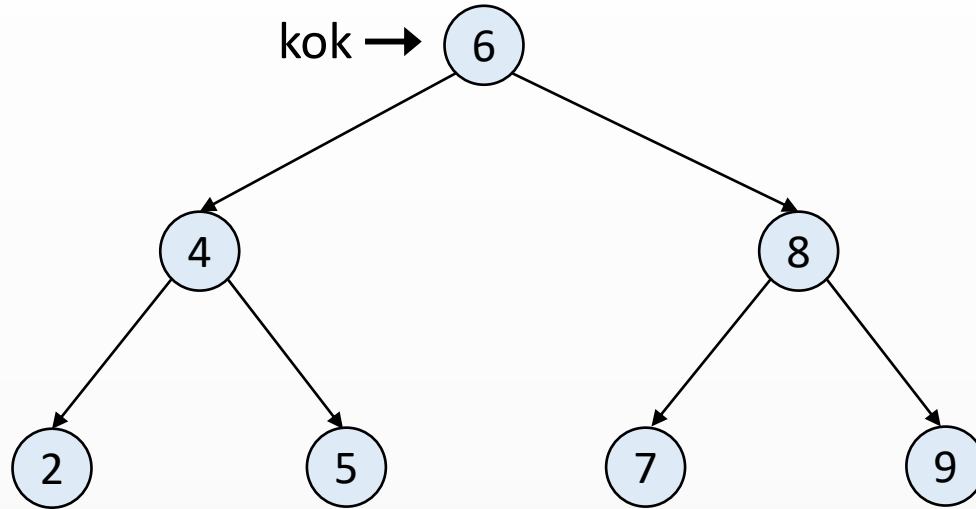


metot çağırımı

satır no

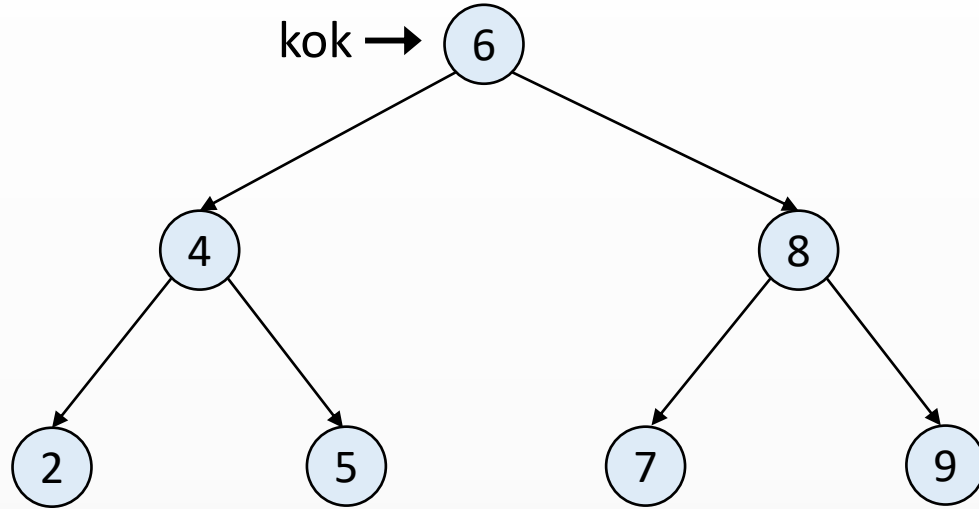
kok

anahtar



anahtar = 5

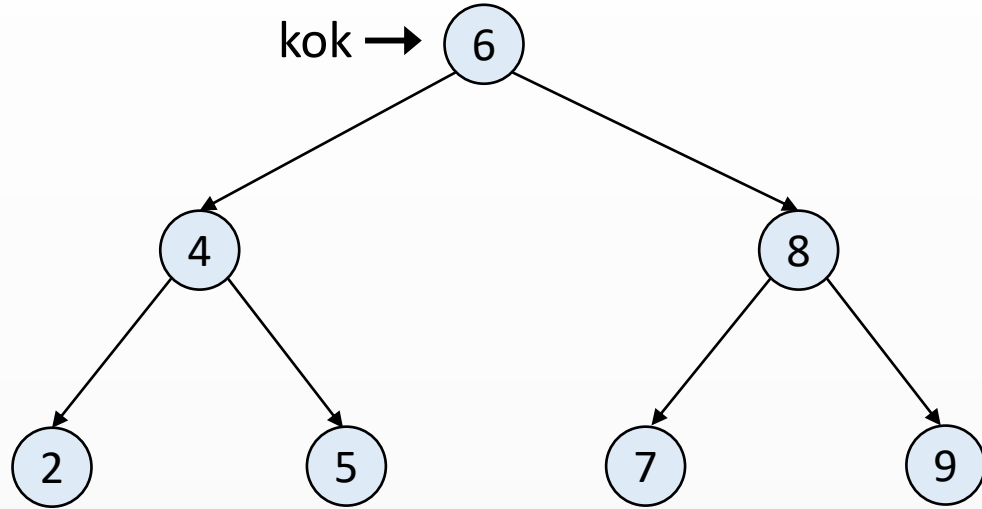
```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



anahtar = 5

metot çağırımı	satır no	kok	anahtar
ara		6	5

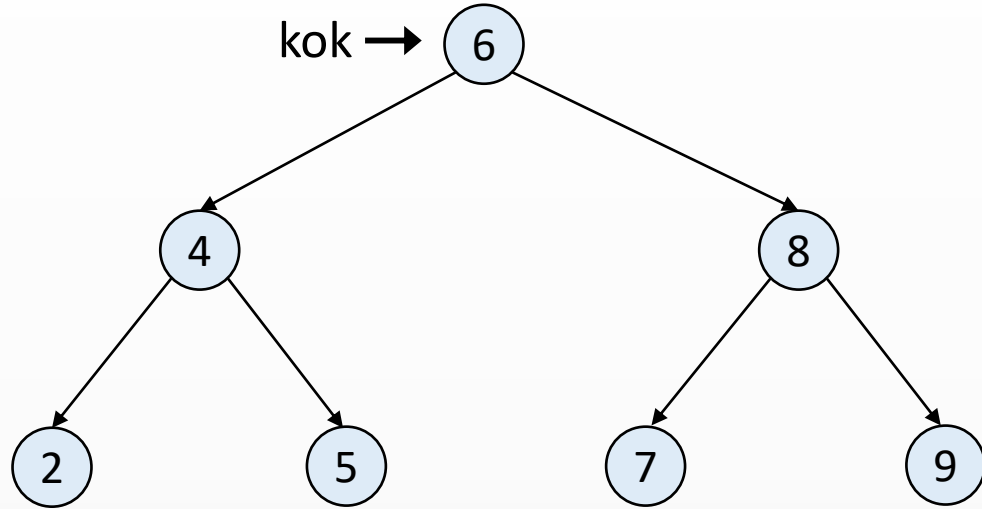
```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



anahtar = 5

metot çağırımı	satır no	kok	anahtar
ara		6	5

```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```

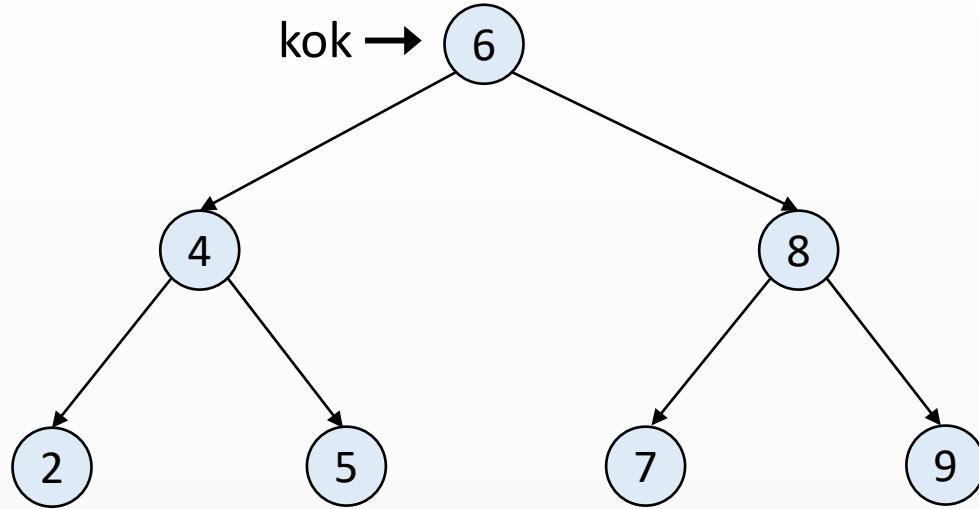


anahtar = 5

metot çağırımı	satır no	kok	anahtar
ara		6	5



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```

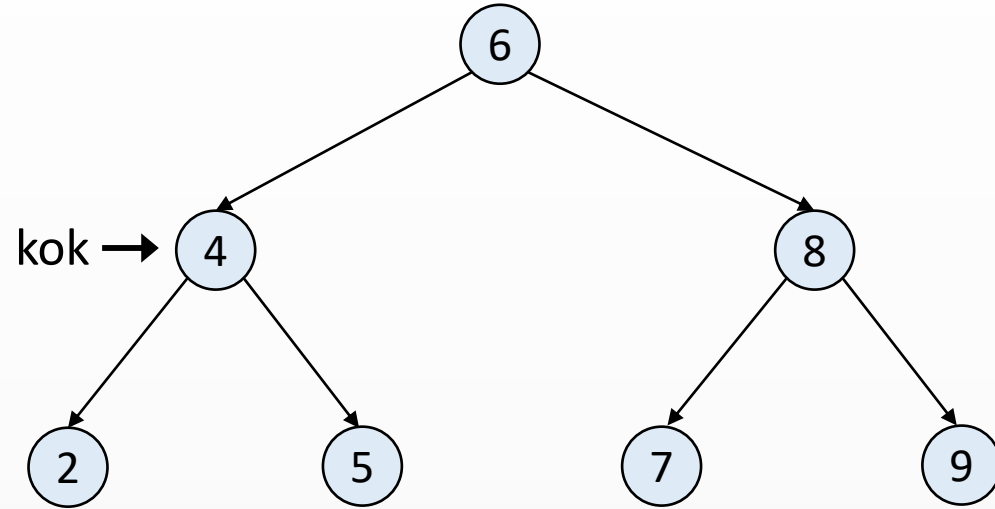


anahtar = 5

metot çağırımı	satır no	kok	anahtar
ara	7	6	5



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```

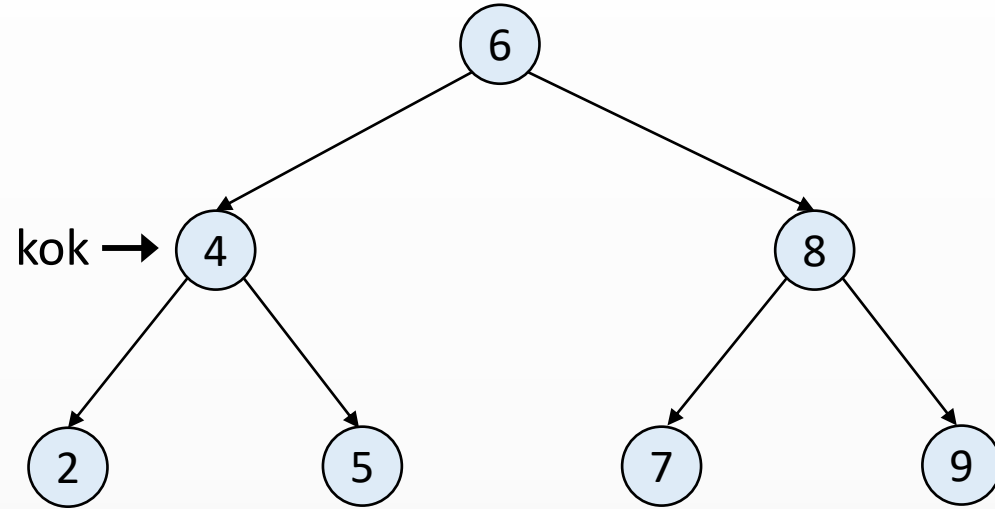
anahtar = 5



metot çağırımı	satır no	kok	anahtar
ara	7	6	5
ara		4	5



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



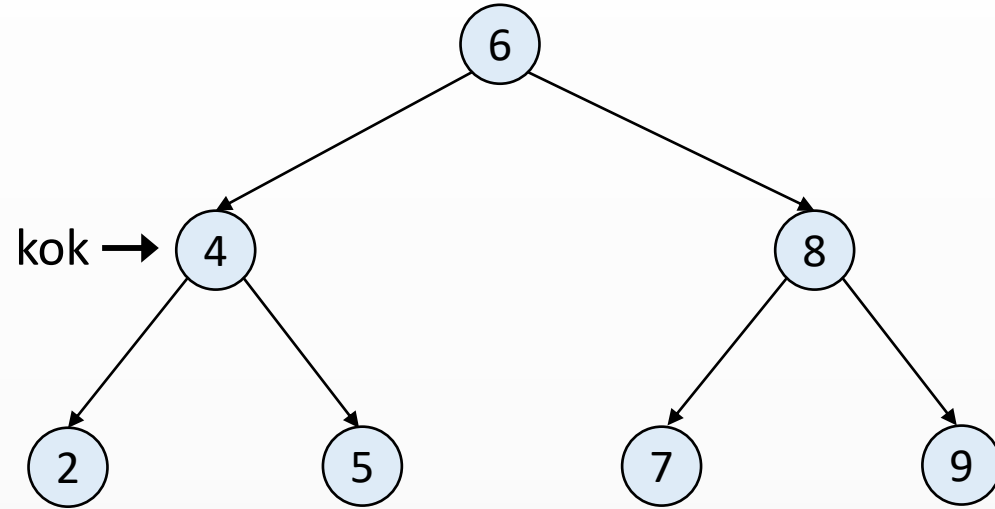
anahtar = 5



metot çağırımı	satır no	kok	anahtar
ara	7	6	5
ara		4	5



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



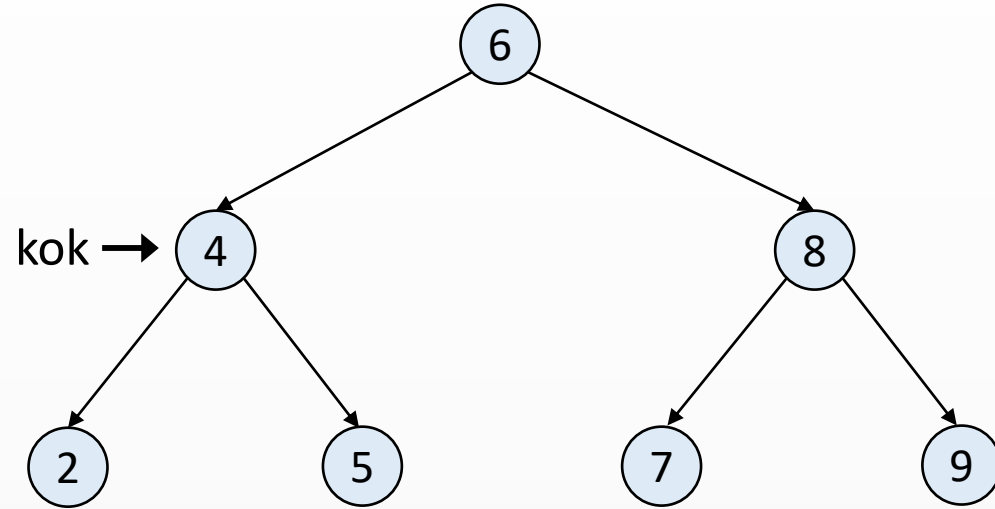
anahtar = 5



metot çağırımı	satır no	kok	anahtar
ara	7	6	5
ara		4	5



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



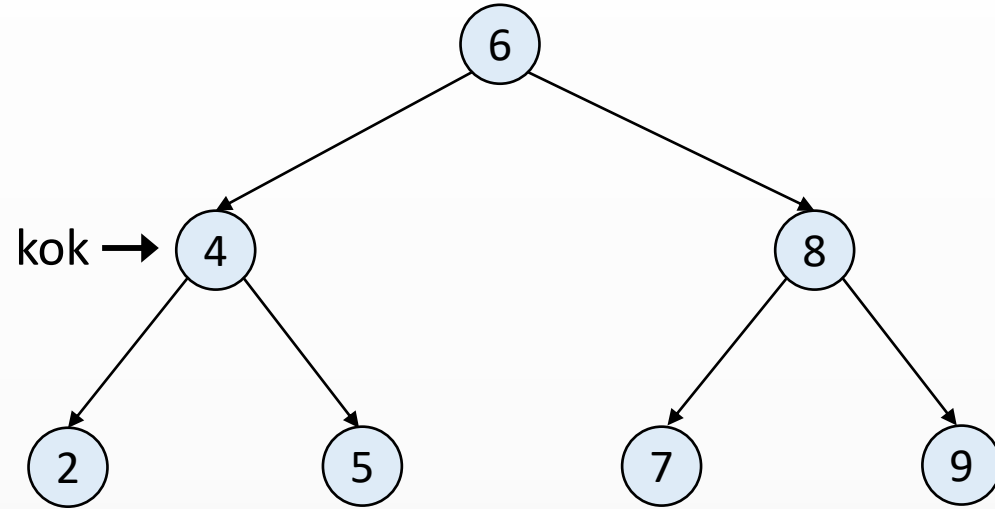
anahtar = 5



metot çağırımı	satır no	kok	anahtar
ara	7	6	5
ara		4	5



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



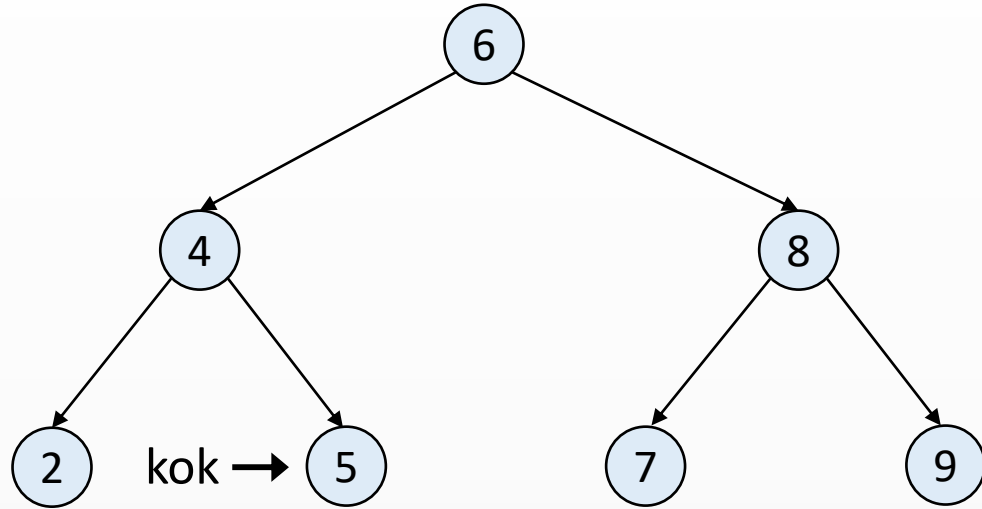
anahtar = 5



metot çağırımı	satır no	kok	anahtar
ara	7	6	5
ara	9	4	5



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



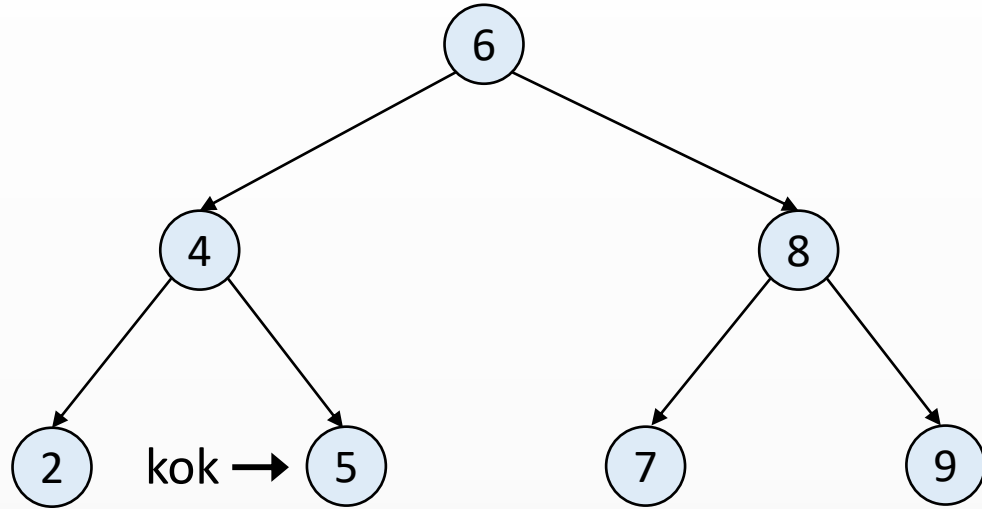
anahtar = 5



metot çağırımı	satır no	kok	anahtar
ara	7	6	5
ara	9	4	5
ara		5	5



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



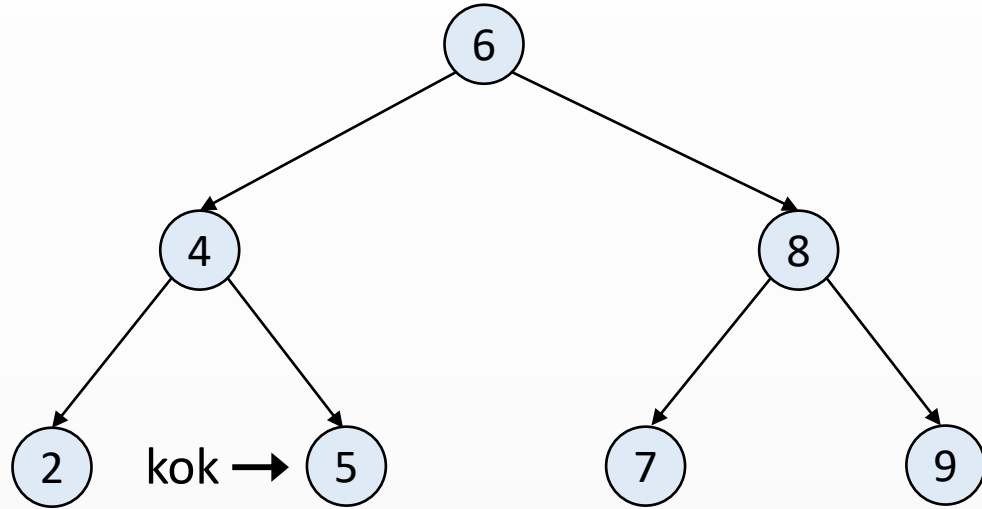
anahtar = 5



metot çağırımı	satır no	kok	anahtar
ara	7	6	5
ara	9	4	5
ara		5	5



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



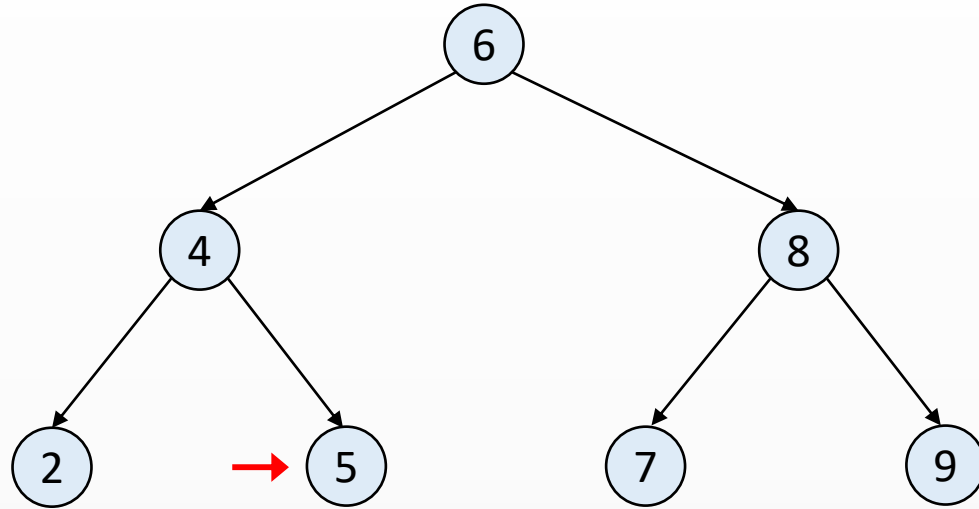
anahtar = 5



metot çağırımı	satır no	kok	anahtar
ara	7	6	5
ara	9	4	5
ara		5	5



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```

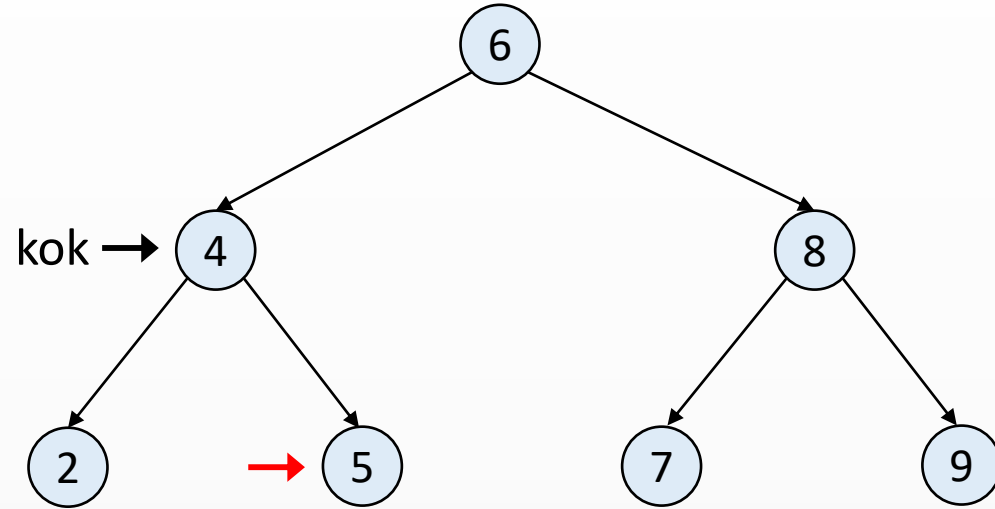



anahtar = 5



metot çağırımı	satır no	kok	anahtar
ara	7	6	5
ara	9	4	5

```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



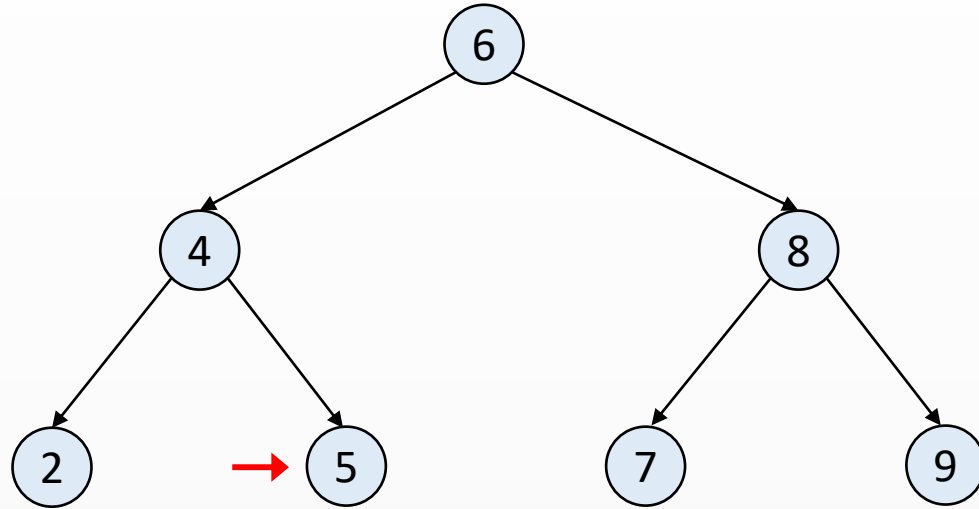
anahtar = 5



metot çağırımı	satır no	kok	anahtar
ara	7	6	5
ara	9	4	5



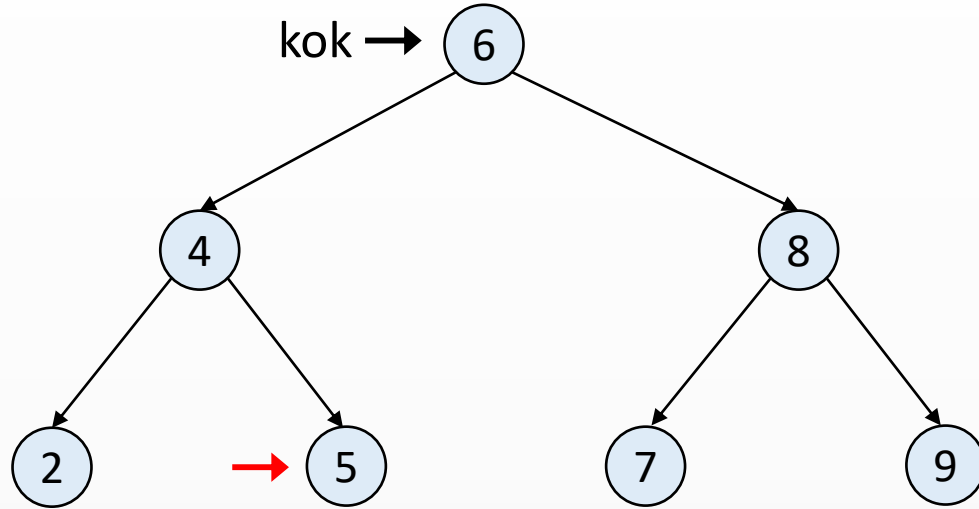
```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



anahtar = 5

metot çağırımı	satır no	kok	anahtar
ara	7	6	5

```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



anahtar = 5

metot çağırımı	satır no	kok	anahtar
ara	7	6	5



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```

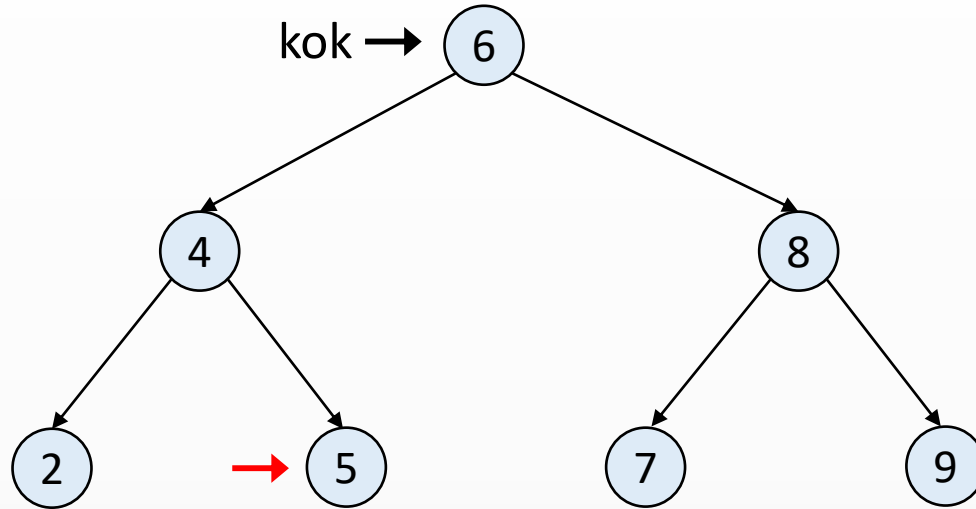


metot çağırımı

satır no

kok

anahtar



anahtar = 5

```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```

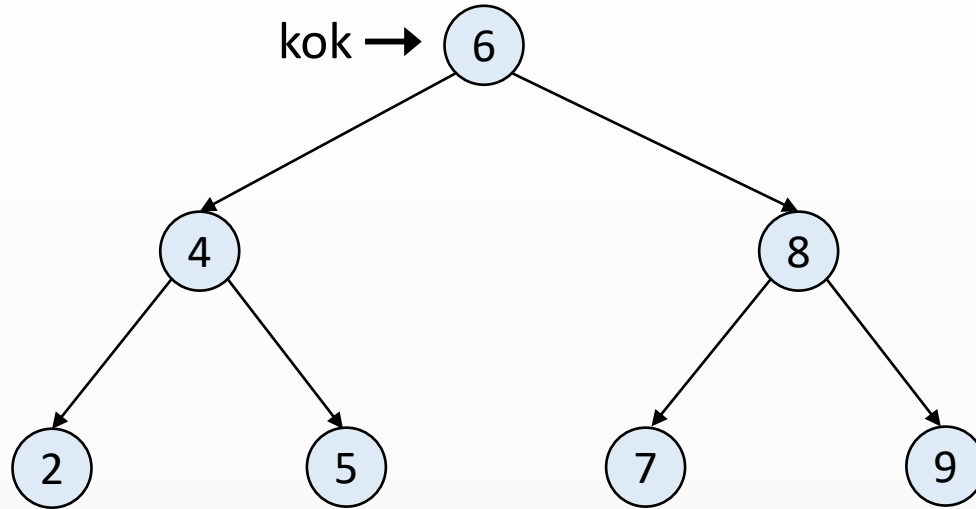


metot çağırımı

satır no

kok

anahtar



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```

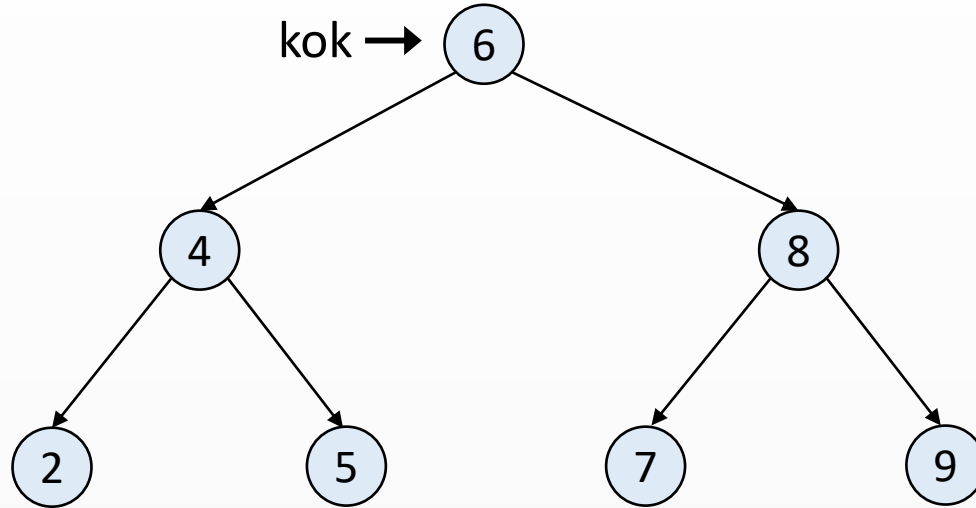


metot çağırımı

satır no

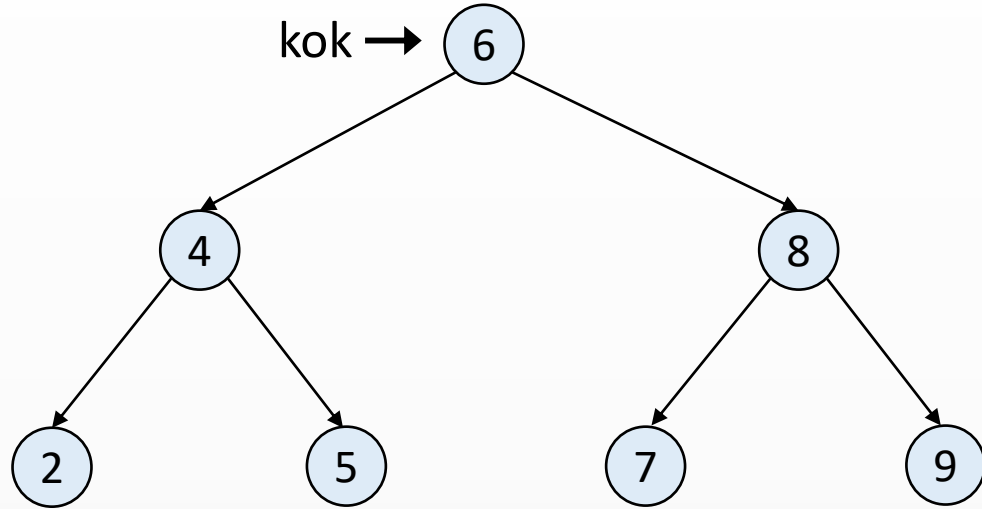
kok

anahtar



anahtar = 10

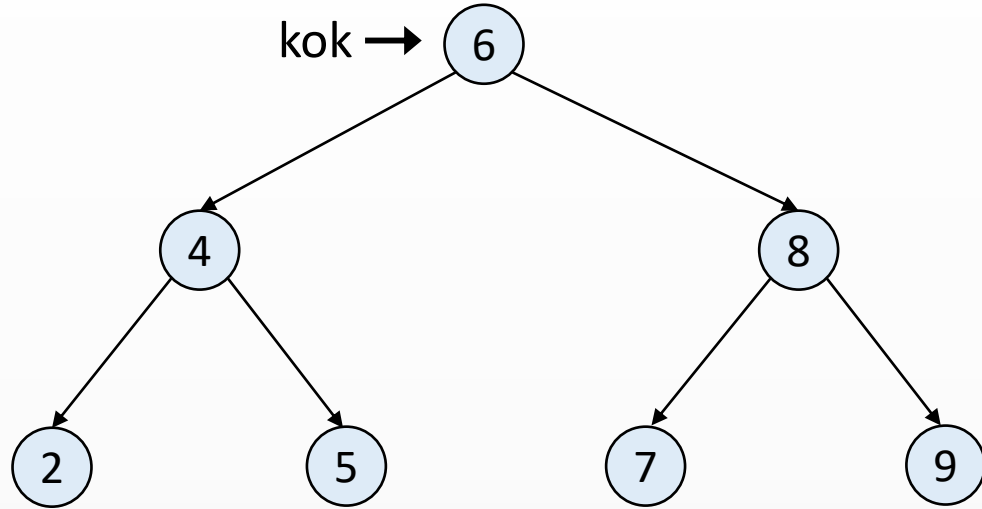
```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



anahtar = 10

metot çağırımı	satır no	kok	anahtar
ara		6	10

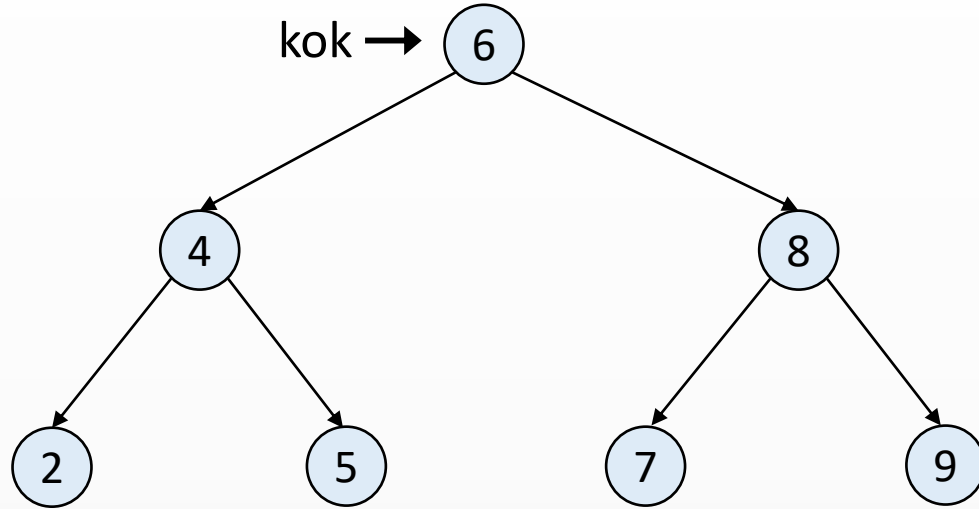
```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```

anahtar = 10

metot çağırımı	satır no	kok	anahtar
ara		6	10

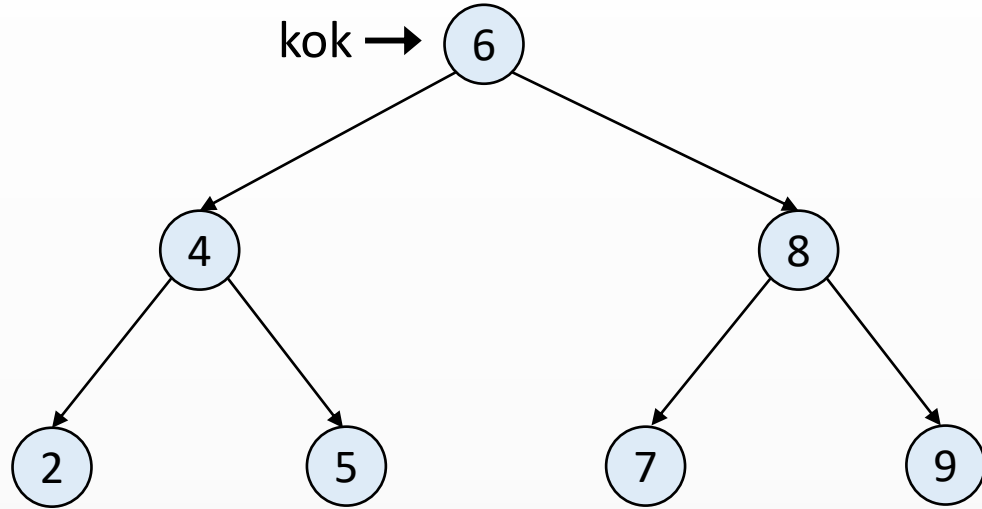
```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



anahtar = 10

metot çağırımı	satır no	kok	anahtar
ara		6	10

```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```

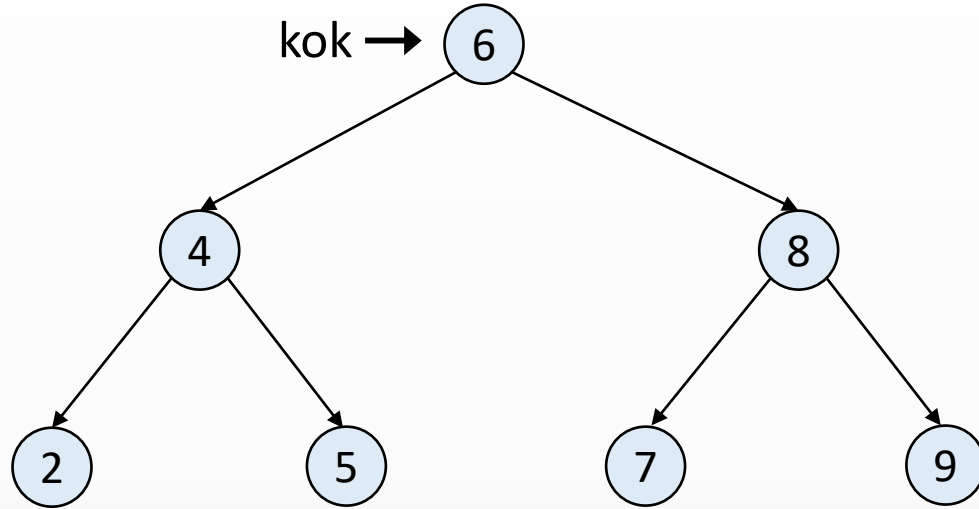


anahtar = 10

metot çağırımı	satır no	kok	anahtar
ara		6	10



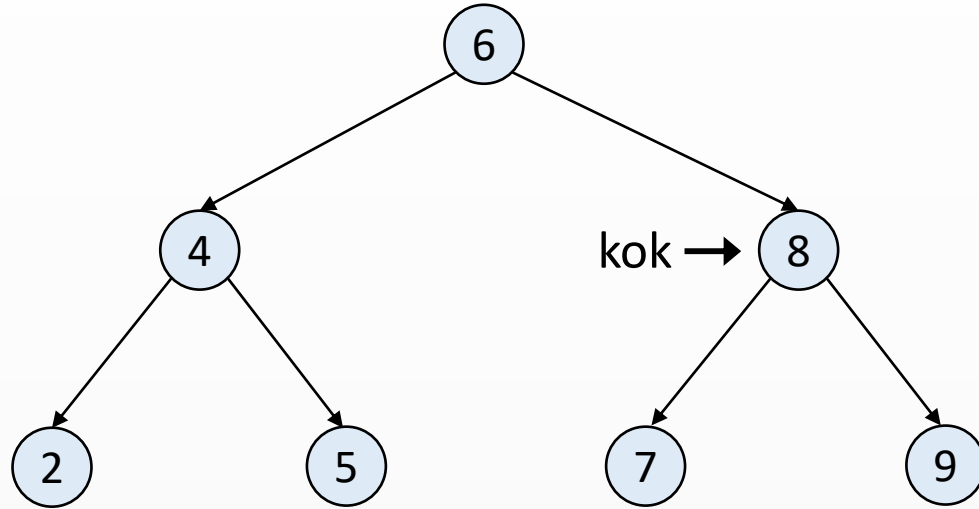
```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



anahtar = 10

metot çağırımı	satır no	kok	anahtar
→ ara	9	6	10

```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



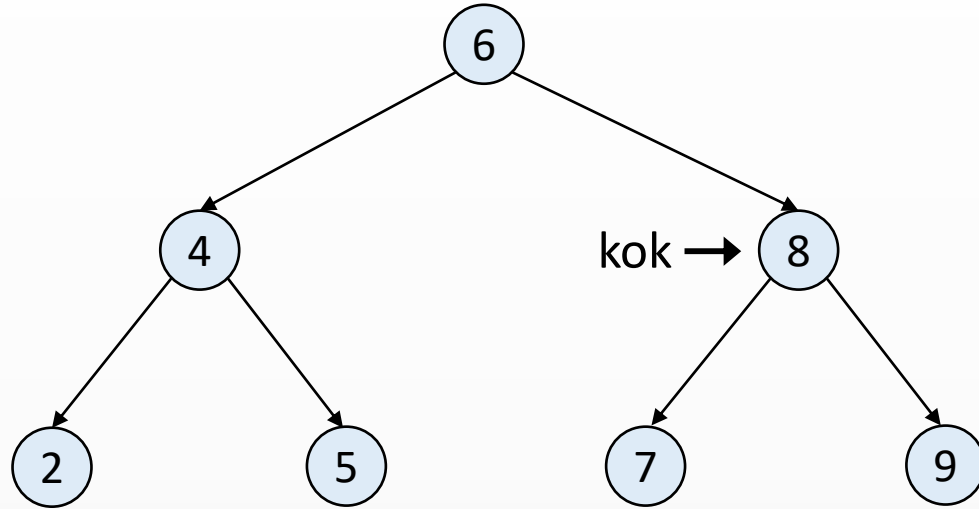
anahtar = 10



metot çağırımı	satır no	kok	anahtar
ara	9	6	10
ara		8	10



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



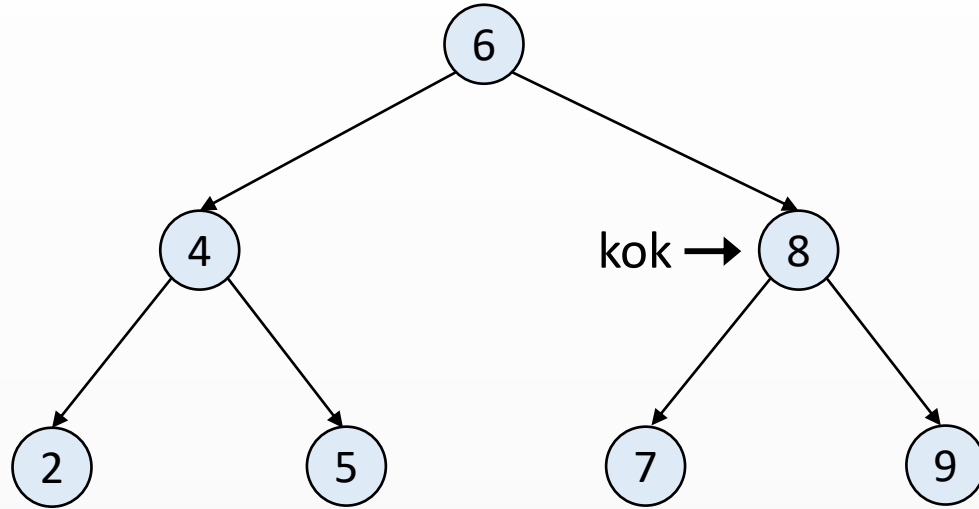
anahtar = 10



metot çağırımı	satır no	kok	anahtar
ara	9	6	10
ara		8	10



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



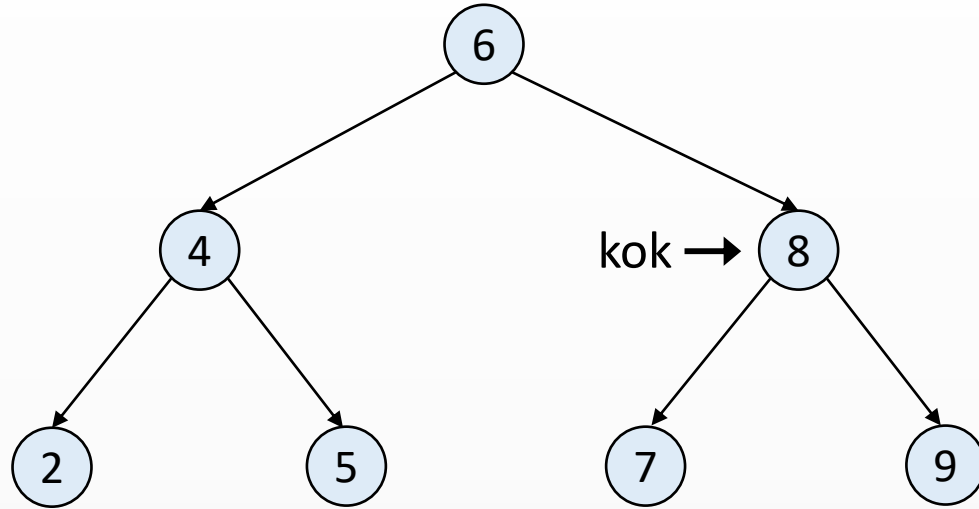
anahtar = 10



metot çağırımı	satır no	kok	anahtar
ara	9	6	10
ara		8	10



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



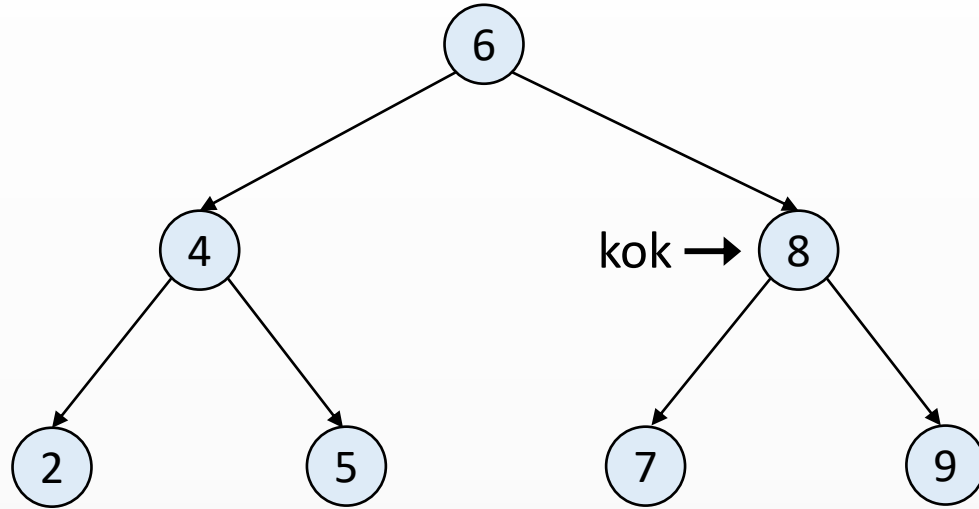
anahtar = 10



metot çağırımı	satır no	kok	anahtar
ara	9	6	10
ara		8	10



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```

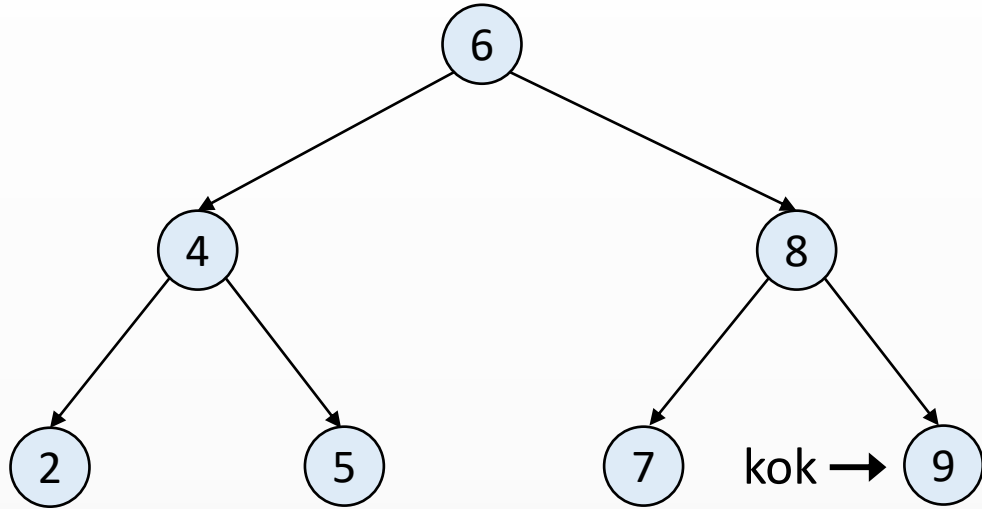
anahtar = 10



metot çağırımı	satır no	kok	anahtar
ara	9	6	10
ara	9	8	10



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



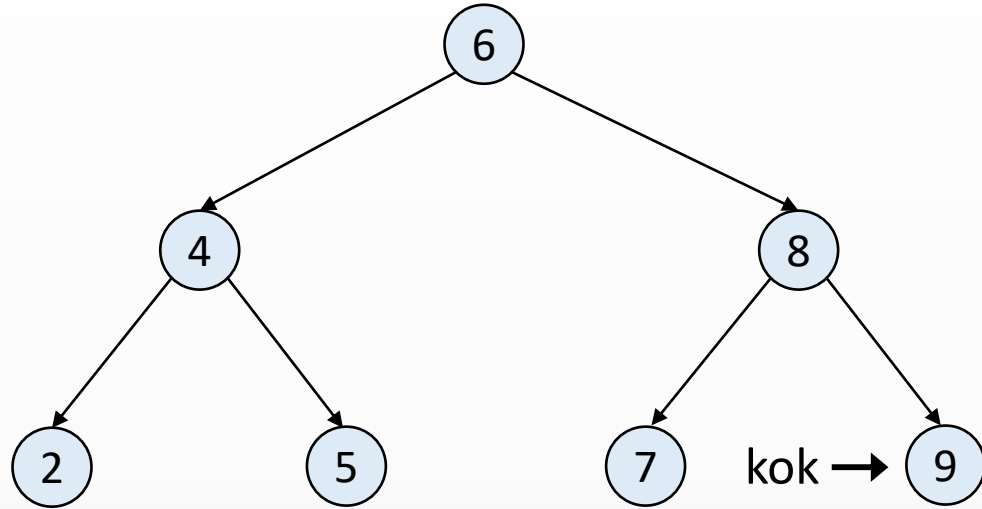
anahtar = 10



metot çağırımı	satır no	kok	anahtar
ara	9	6	10
ara	9	8	10
ara		9	10



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



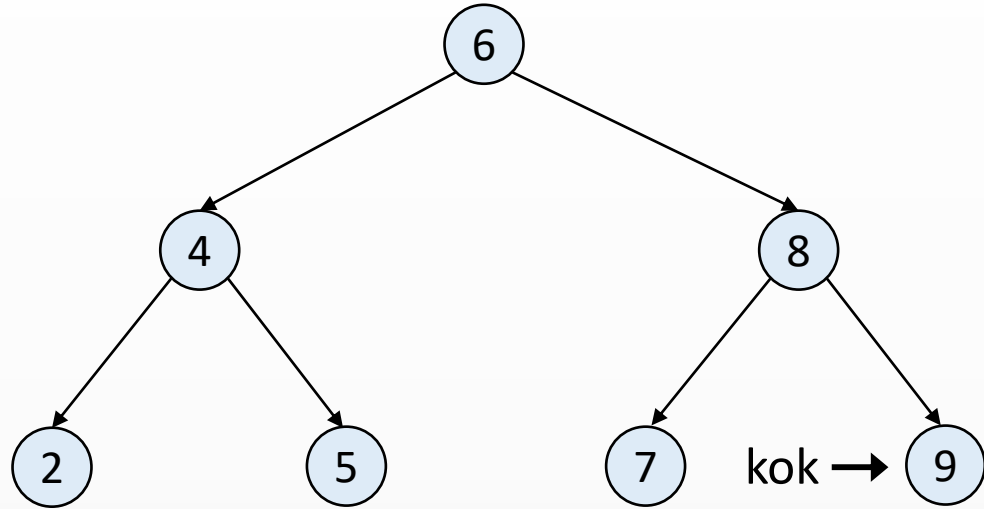
anahtar = 10



metot çağırımı	satır no	kok	anahtar
ara	9	6	10
ara	9	8	10
ara		9	10



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



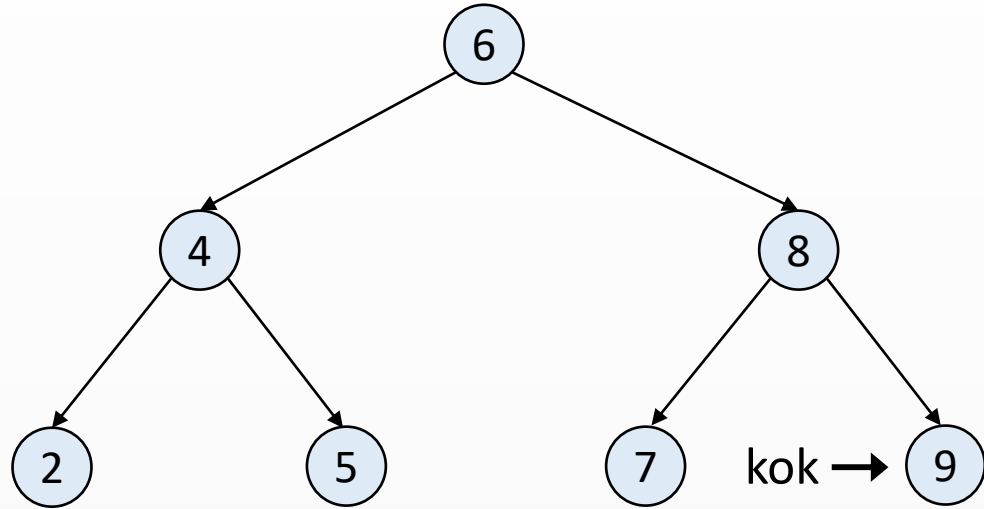
anahtar = 10



metot çağırımı	satır no	kok	anahtar
ara	9	6	10
ara	9	8	10
ara		9	10



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



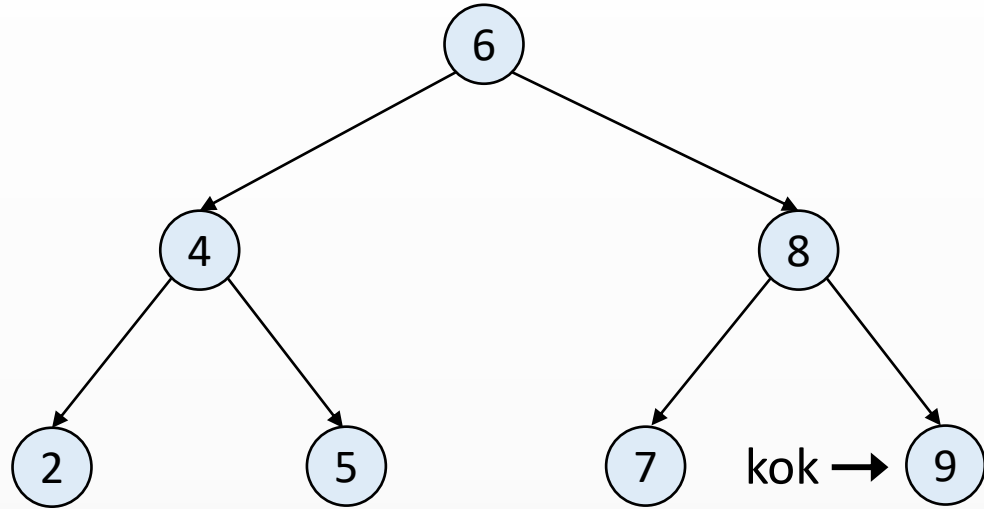
anahtar = 10



metot çağırımı	satır no	kok	anahtar
ara	9	6	10
ara	9	8	10
ara		9	10



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



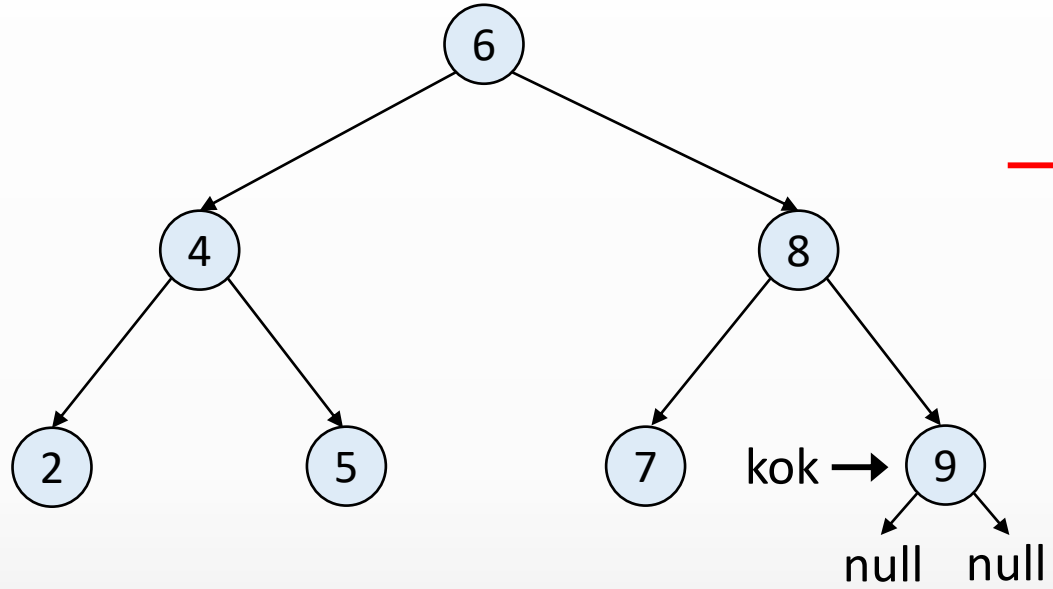
anahtar = 10



metot çağırımı	satır no	kok	anahtar
ara	9	6	10
ara	9	8	10
ara	9	9	10



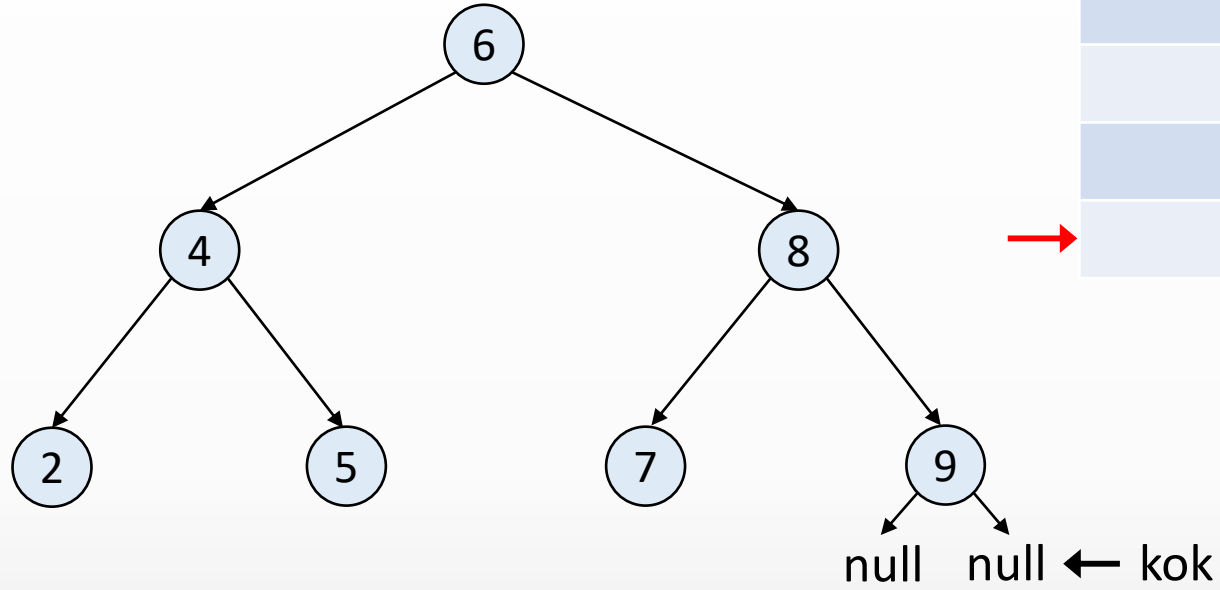
```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



metot çağırımı	satır no	kok	anahtar
ara	9	6	10
ara	9	8	10
ara	9	9	10

anahtar = 10

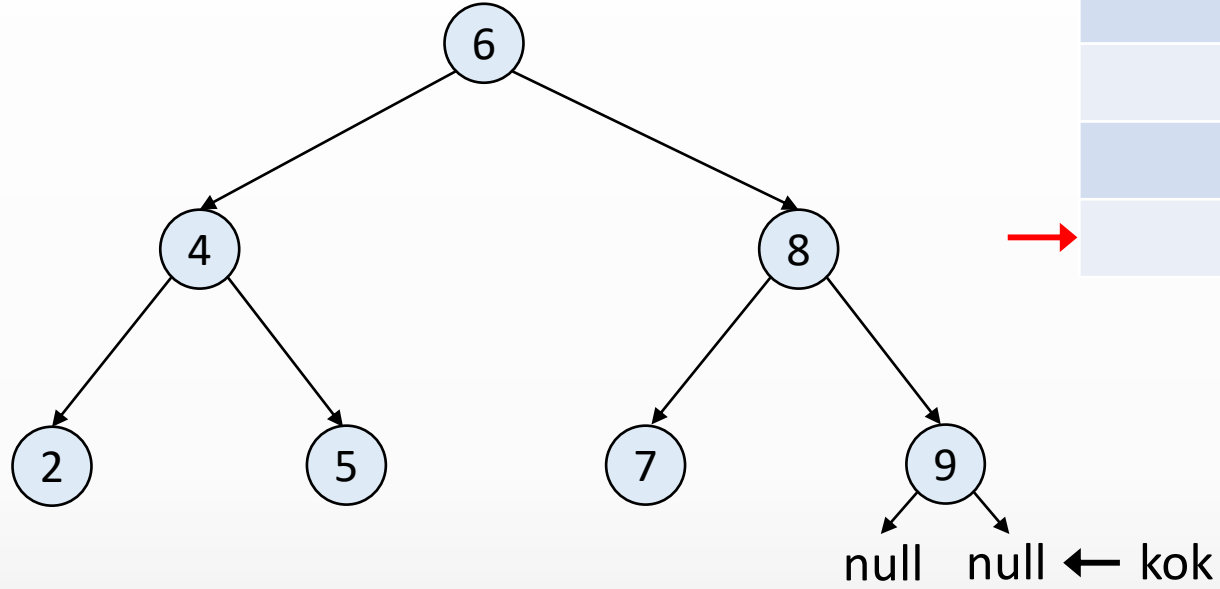
```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



metot çağırımı	satır no	kok	anahtar
ara	9	6	10
ara	9	8	10
ara	9	9	10
ara		null	10

anahtar = 10

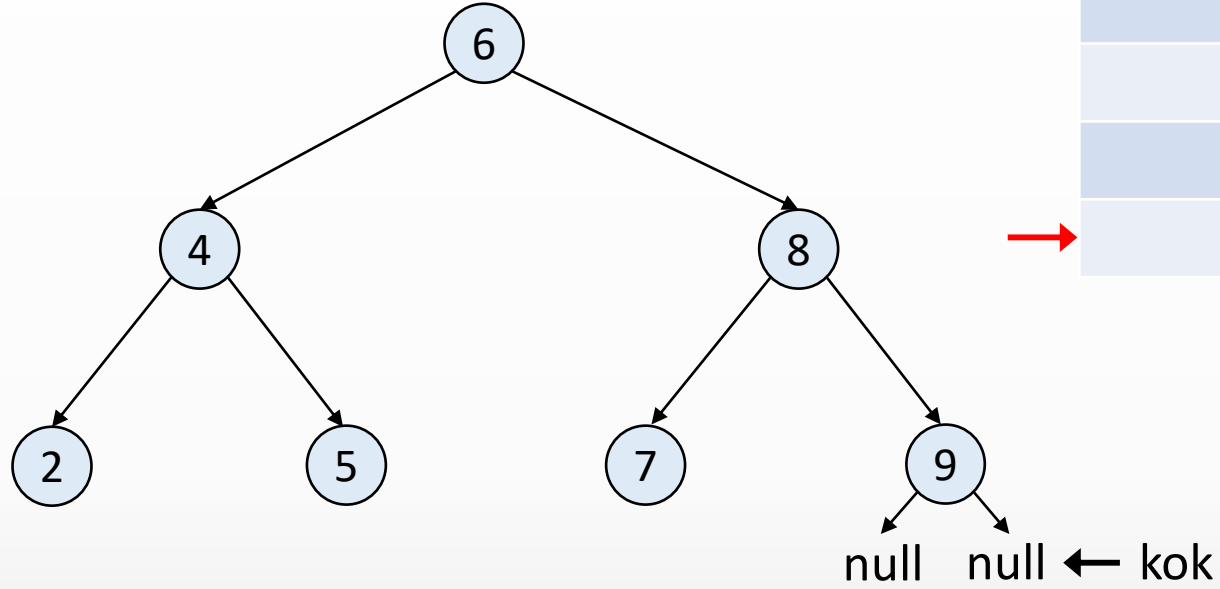
```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```

metot çağırımı	satır no	kok	anahtar
ara	9	6	10
ara	9	8	10
ara	9	9	10
ara		null	10

anahtar = 10

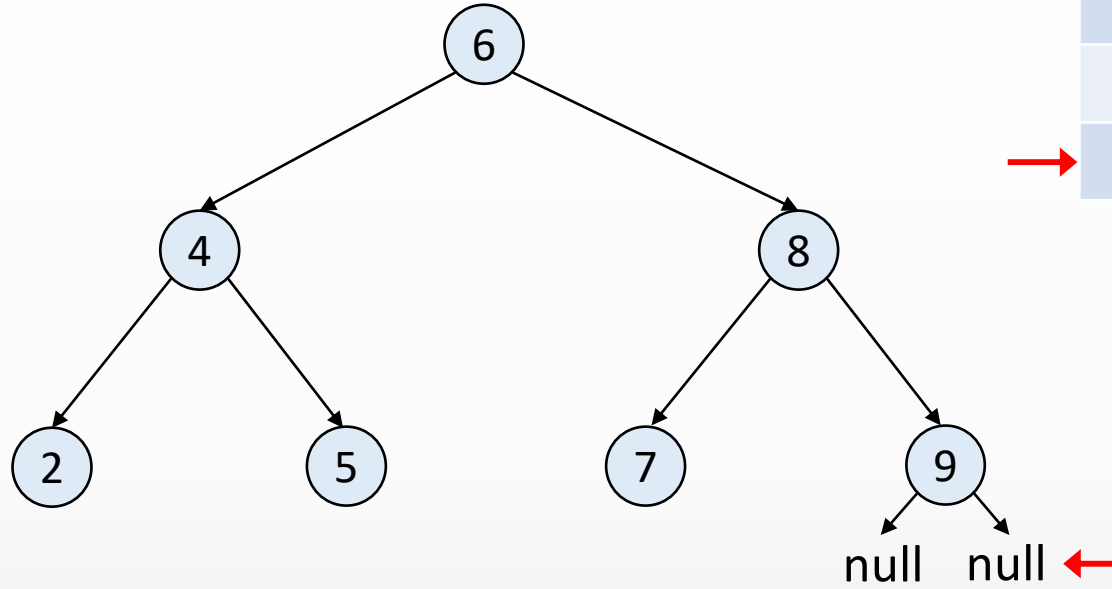
```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



metot çağırımı	satır no	kok	anahtar
ara	9	6	10
ara	9	8	10
ara	9	9	10
ara		null	10

anahtar = 10

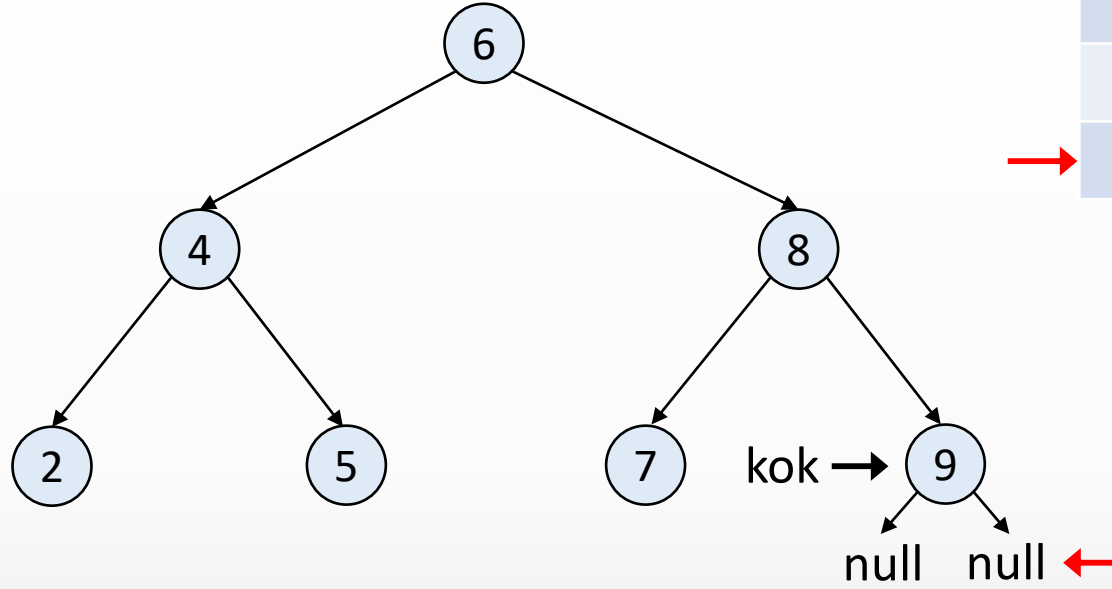
```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



metot çağırımı	satır no	kok	anahtar
ara	9	6	10
ara	9	8	10
ara	9	9	10

anahtar = 10

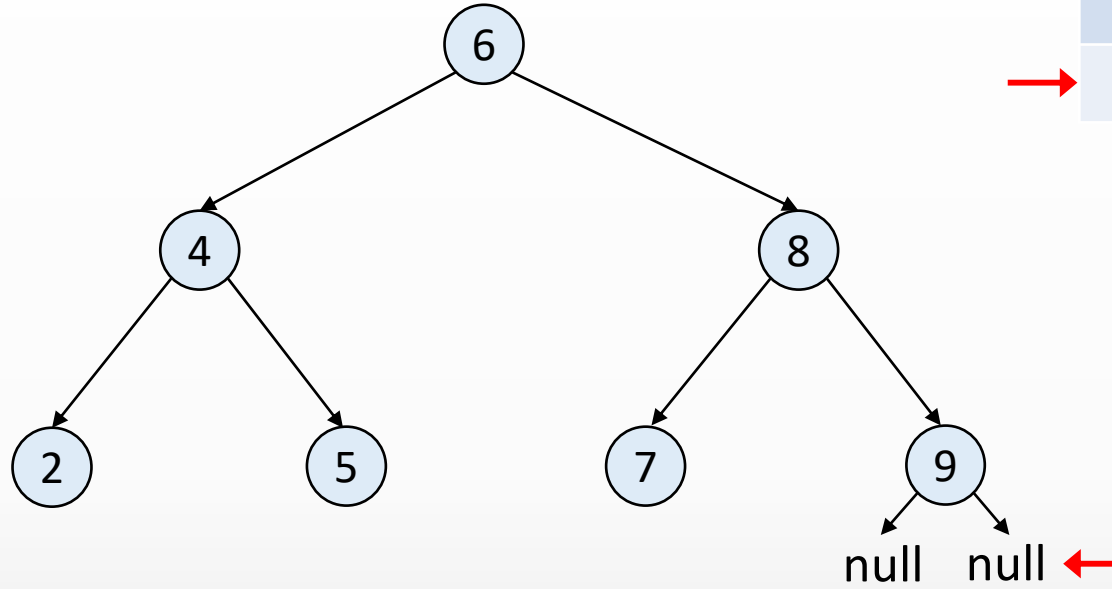
```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



metot çağırımı	satır no	kok	anahtar
ara	9	6	10
ara	9	8	10
ara	9	9	10

anahtar = 10

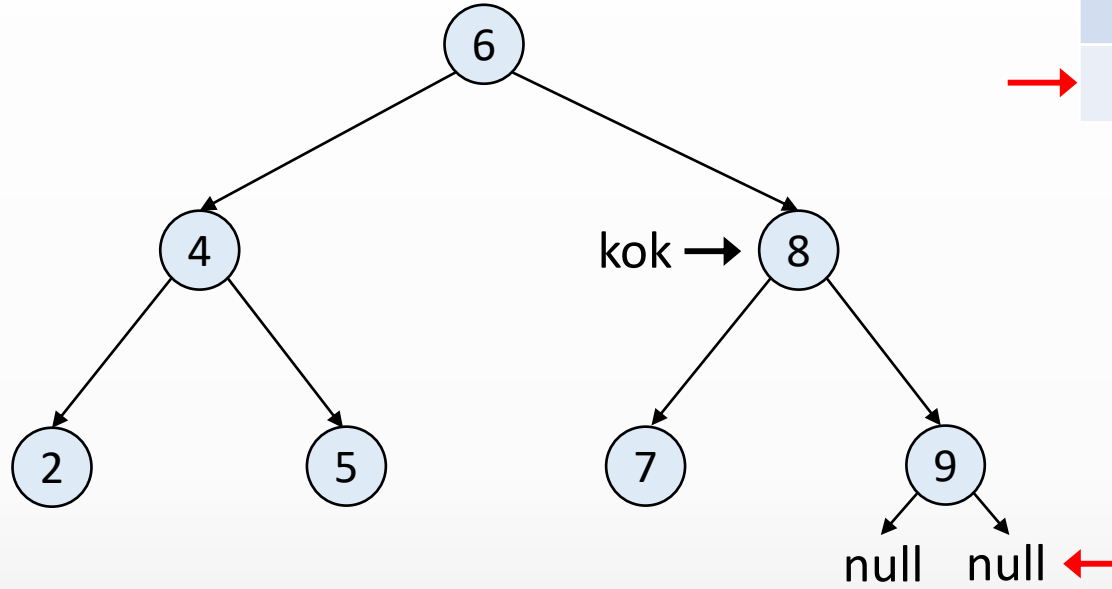
```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



metot çağırımı	satır no	kok	anahtar
ara	9	6	10
ara	9	8	10

anahtar = 10

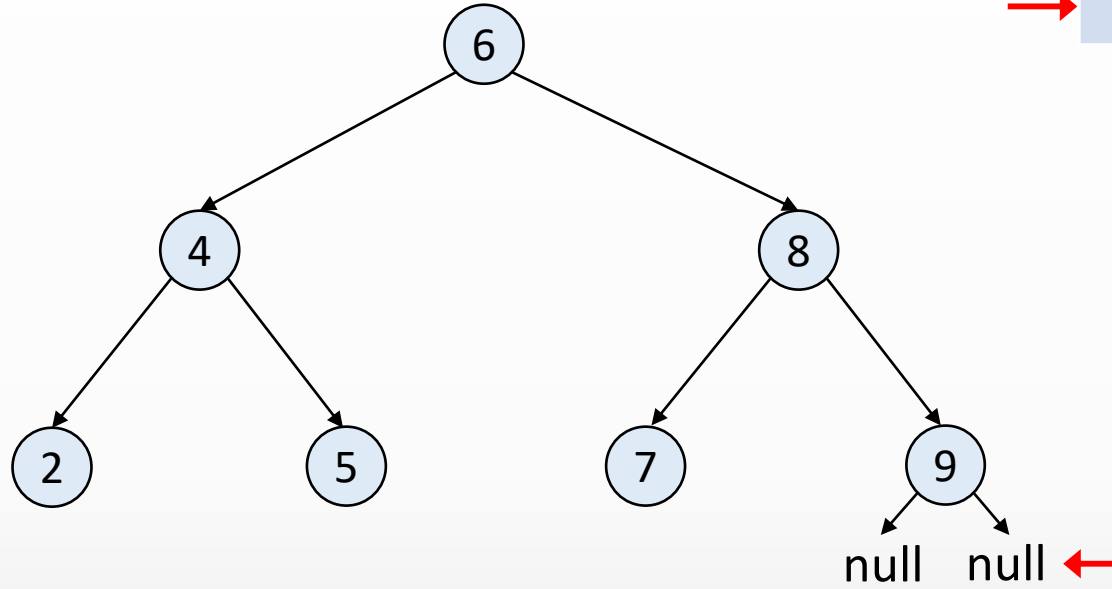
```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



metot çağırımı	satır no	kok	anahtar
ara	9	6	10
ara	9	8	10

anahtar = 10

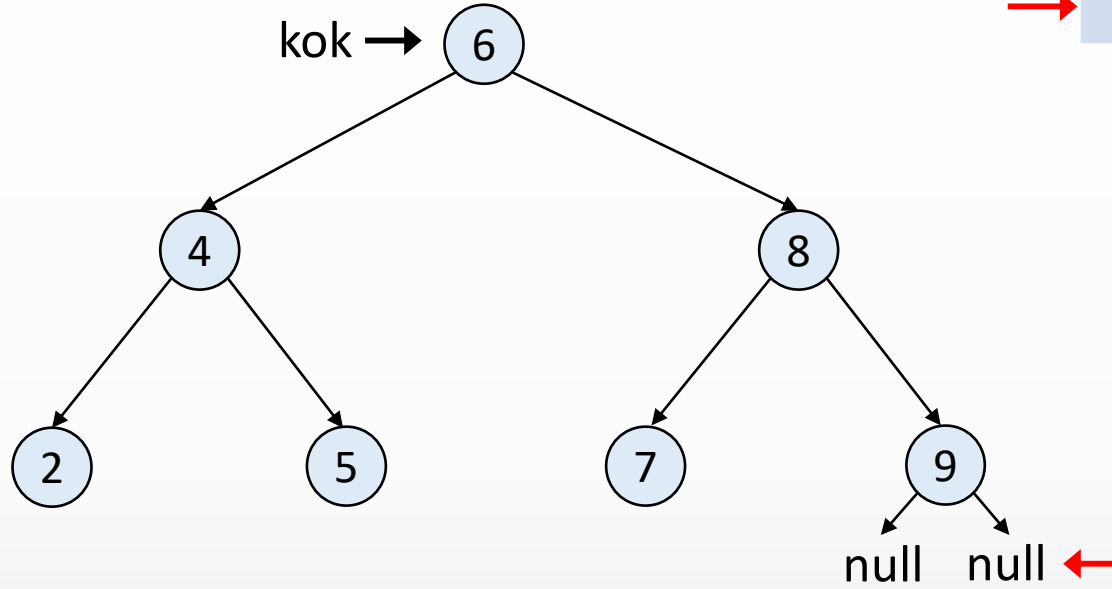
```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



anahtar = 10

metot çağırımı	satır no	kok	anahtar
ara	9	6	10

```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```



metot çağırımı	satır no	kok	anahtar
ara	9	6	10

anahtar = 10

```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```

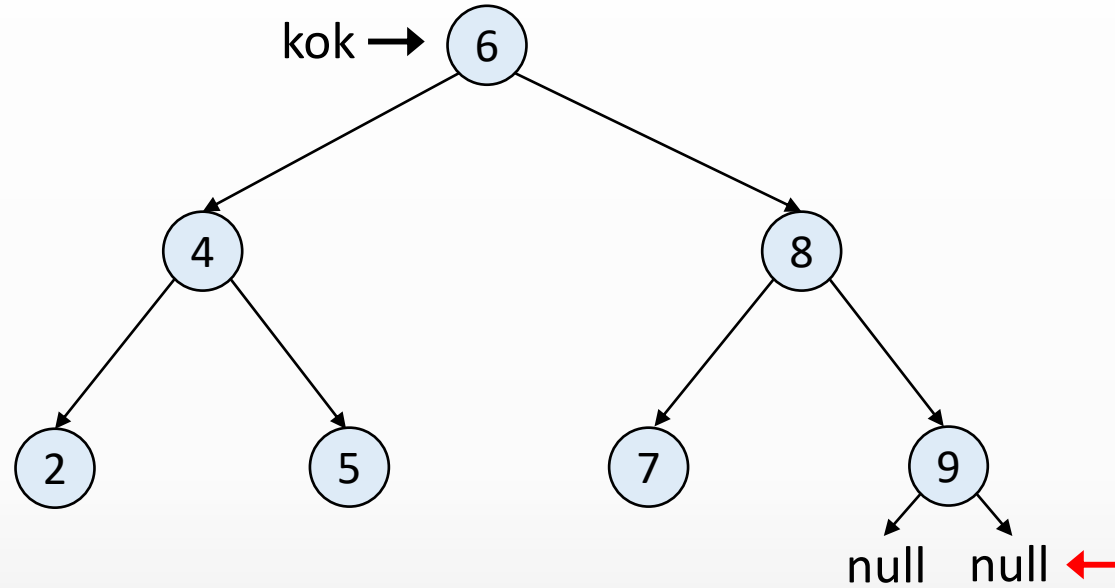



metot çağırımı

satır no

kok

anahtar



```
1 public AgacDugumu ara(AgacDugumu kok, int anahtar)
2 {
3     if(kok == null || kok.veri == anahtar) {
4         return kok;
5     }
6     if(anahtar < kok.veri)
7         return ara(kok.sol, anahtar);
8     else
9         return ara(kok.sag, anahtar);
10 }
```





SON