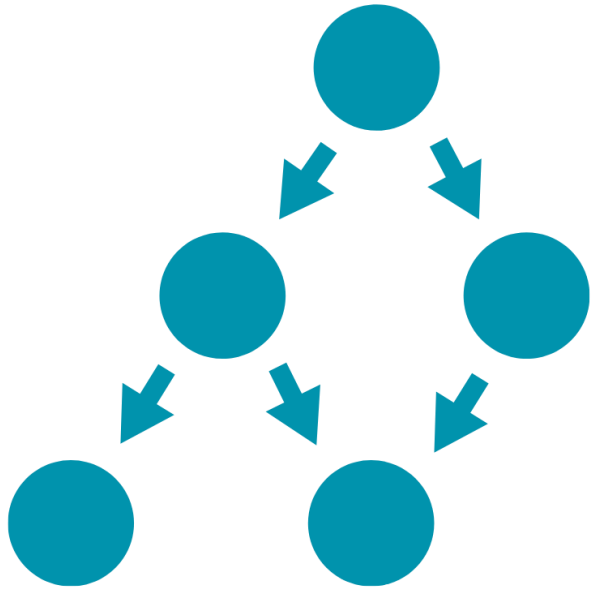
BTK AKADEMİ

ZAFER CÖMERT
Öğretim Üyesi



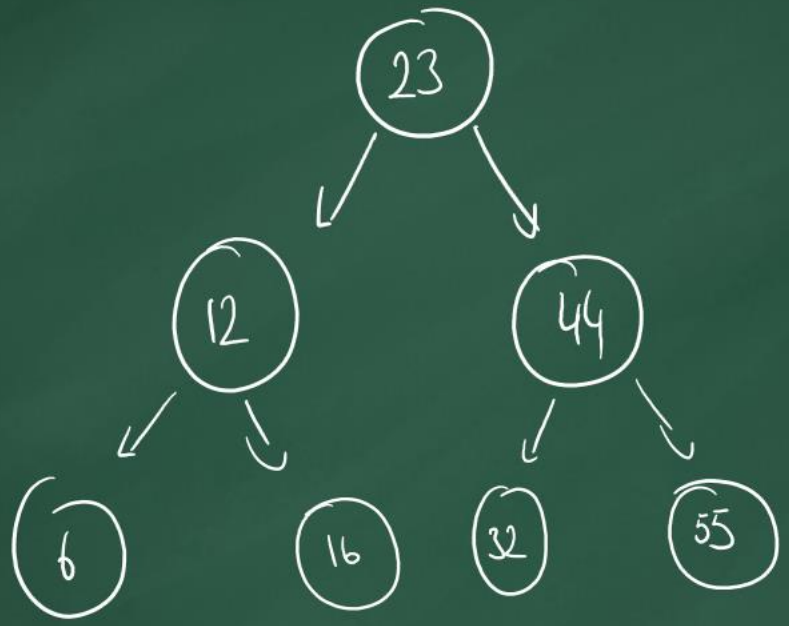
VERİ YAPILARILARI VE ALGORİTMALAR

BST Problems

Giriş

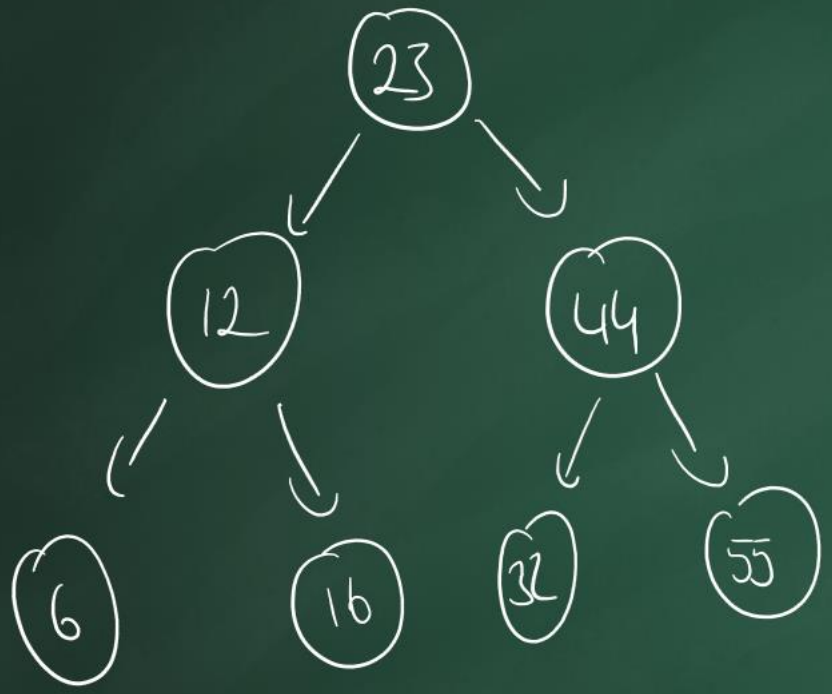
1. Level Order Dolaşma
2. Maksimum derinlik (Maximum depth)
3. Maksimum elemanın bulunması (Maximum item)
4. En derin elemanın bulunması (Deepest node)
5. Yaprak sayısının bulunması (Number of Leaves)
6. Ağaçtaki full-düğüm sayısının bulunması (Number of Full Nodes)
7. Ağaçtaki yarım-düğüm sayısının bulunması (Number of Half Nodes)
8. Bir ağacın genişliği hesaplayan metot tasarımı (Width of tree)
9. Yığın ile PreOrder Traversal (PreOrder Traversal with Stack)
10. Kökten yapraklara olan yolları yazdıran metot tasarımı (Paths)
11. Bir ağacın simetrisini alan metot tasarımı (MirrorOfTree)

Problem : Level Order dolaşma gerçekleştiren metot tasarımı



Output → 23, 12, 44, 6, 16, 32, 55

Level Order Traversal



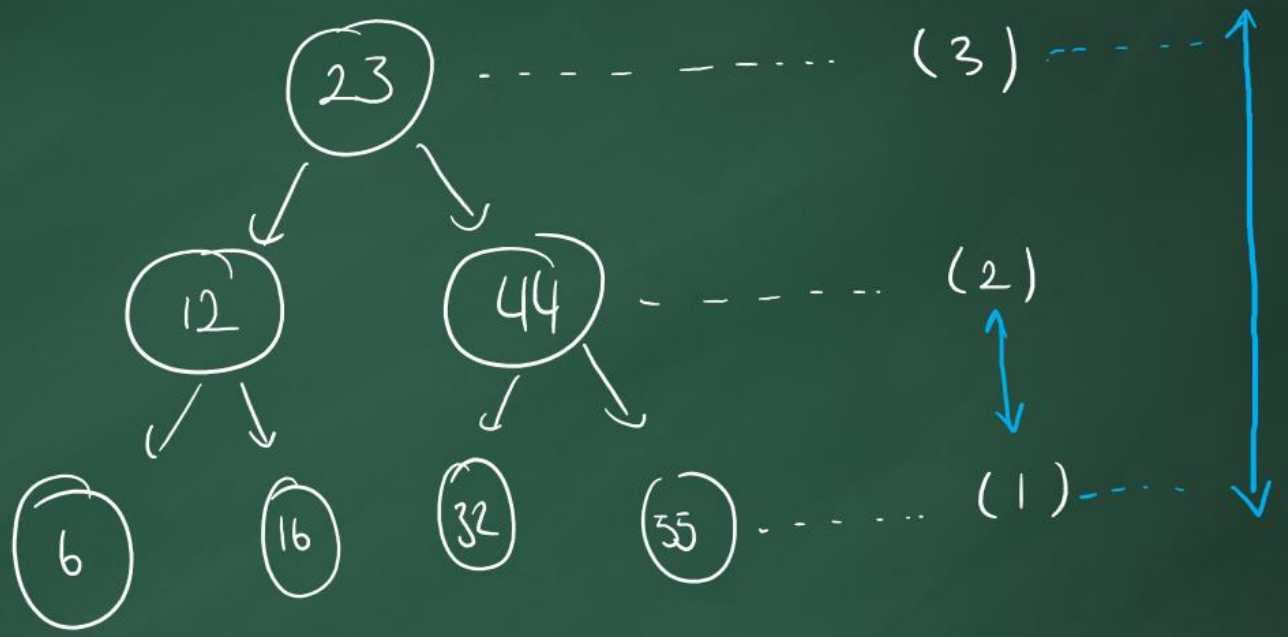
Output

23, 12, 44, 6, 16, 32, 55

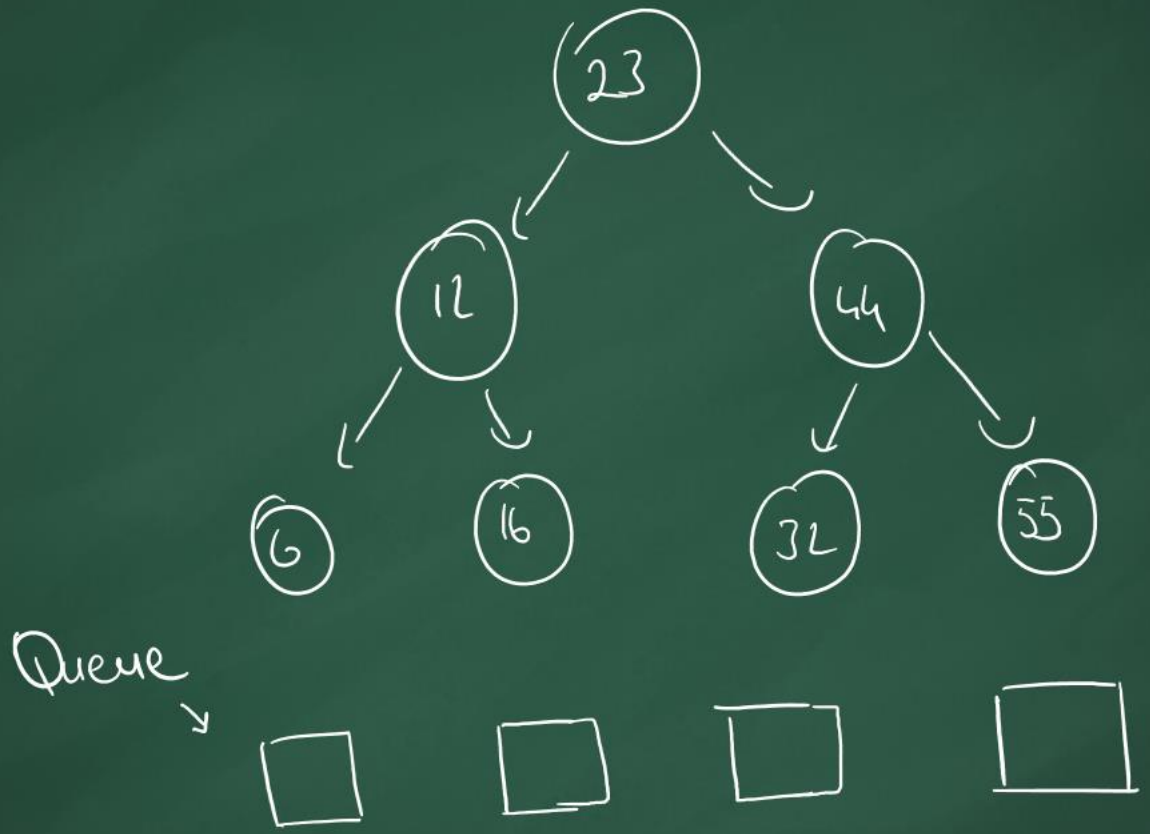
	De Queue
q → 23	23
q → 12 44	12
q → 44 6 16	44
q → 6 16 32 55	6
q → 16 32 55	16
q → 32 55	32
q → 55	55
q → { }	

"first-in first-out – FIFO"
Queue

Problem: Bir ağaçtaki maksimum derinliğin bulunması.



Problem : Maksimum elemanın iteratif olarak bulunması



Problem : Ağaçtaki en derin düğümü bulan metot tasarımı

q → 23

→ 12 44

→ 44 6 16

→ 6 16 32 55

→ 16 32 55 -5

→ 32 55 -5

→ 55 -5

→ -5 100

→ 100

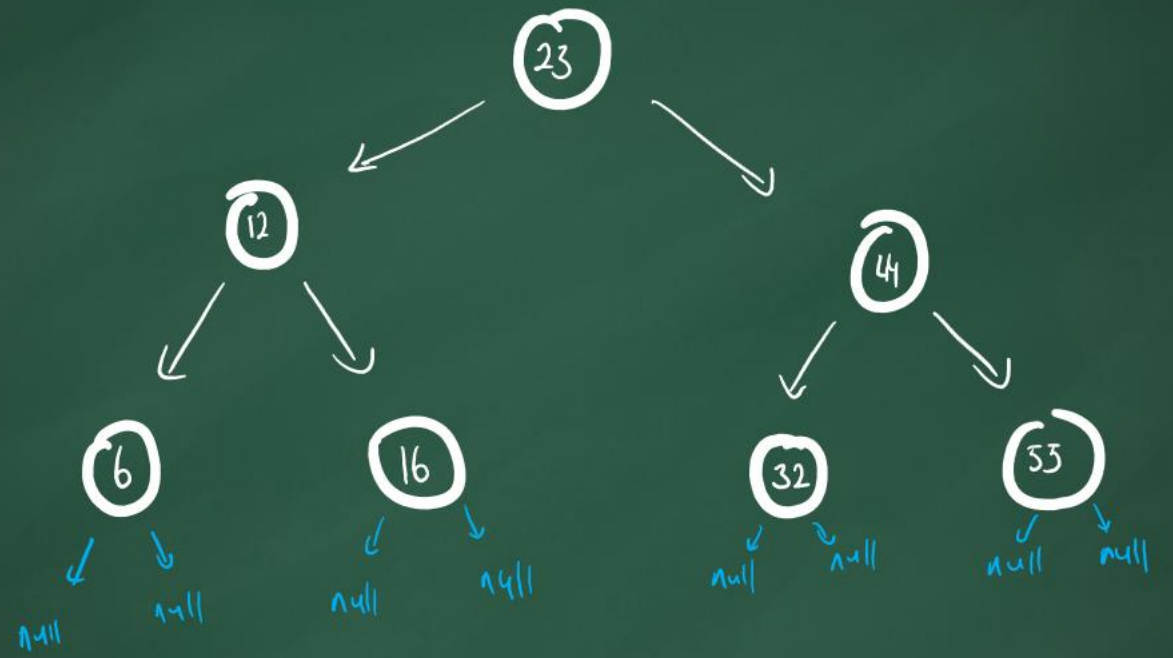
→ 120

base case → return 120

Enqueue

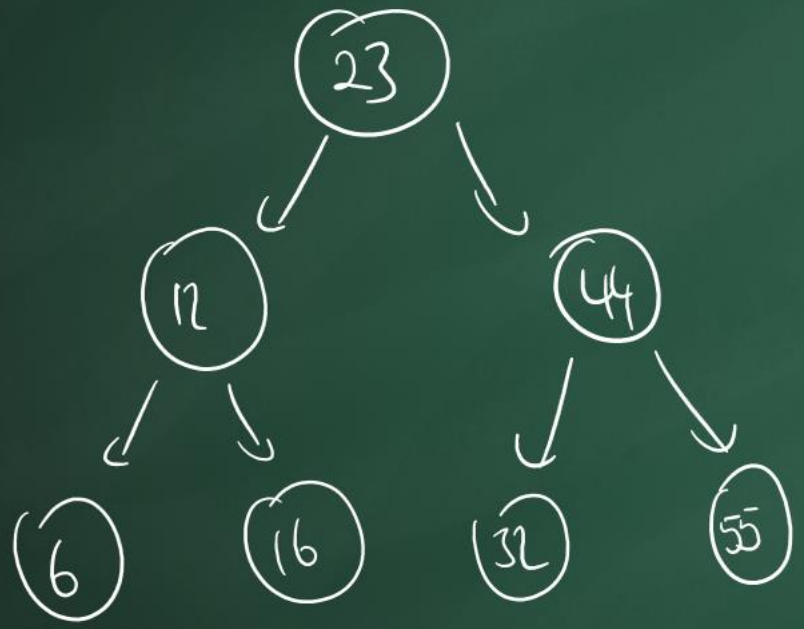
23
12
44
6
16
32
55
-3
100
120

Problem : Ağaçtaki yaprak sayısını bulan metot tasarımı.




```
if (temp.Left = null && temp.Right = null)
    count++
```

Problem → Ağaçtaki full düğüm sayısını verecek metot tasarımı.

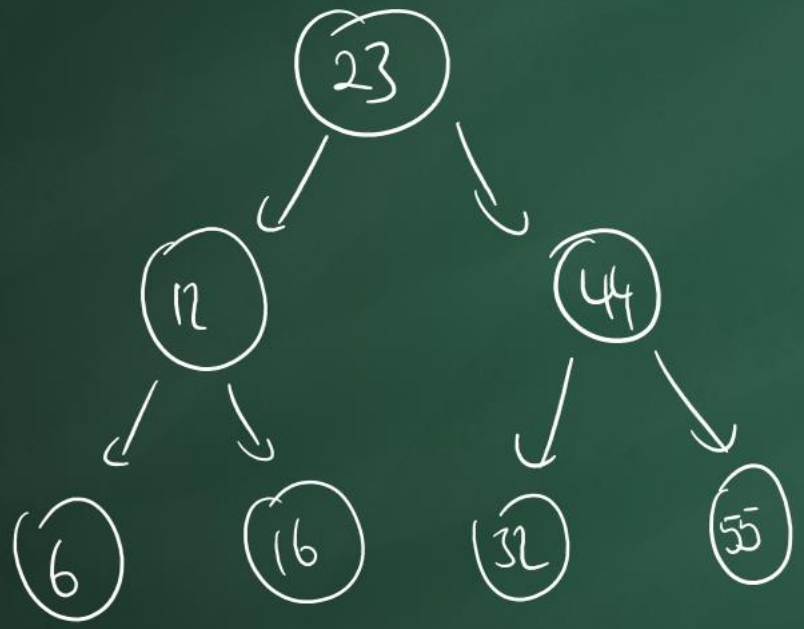


- Level-order traversal
- Mode

```
if (temp.Left != null &&
        temp.Right != null)
    count = count+1
```

Dolaşma sonucunda tam düğüm sayısı elde edilir.

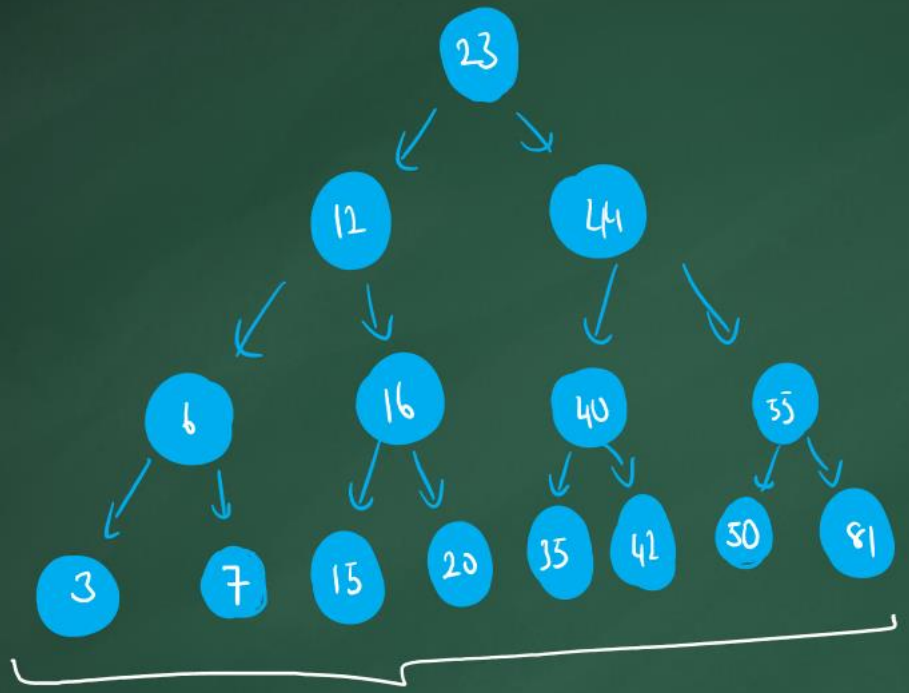
Problem → Ağaçtaki yarım düğüm sayısını verecek metot tasarımı.



- Level-order traversal
- Node
  - temp.Left==null && temp.Right!=null ||
  - temp.Left!=null && temp.Right==null

count=count+1

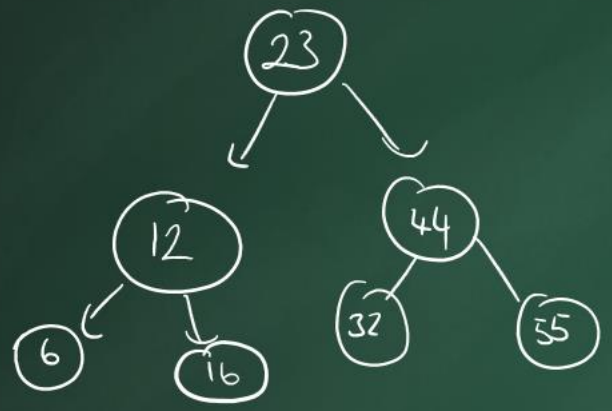
Dolama sonucunda tam düğüm sayısı elde edilir.



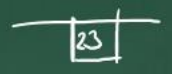
Width of tree (8)

Problem → Bir ağacın genişliğini bulan metot tasarımını gerçekleştiriniz.

PreOrder Traversal with Stack



Stack(S) → S



S → push → 23

while ..... (S.Count > 0)

1 = 23

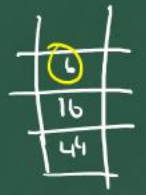
S → pop()

S → push → 44

S → push → 12

2 = 12

S → pop()



S → push → 16

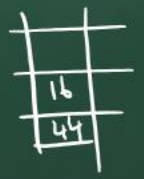
S → push → 6

3 = 6

S → pop()

S → pop()

4 = 16



5 =

S → pop()

S → push → 55

S → push → 32

44

6 =

32

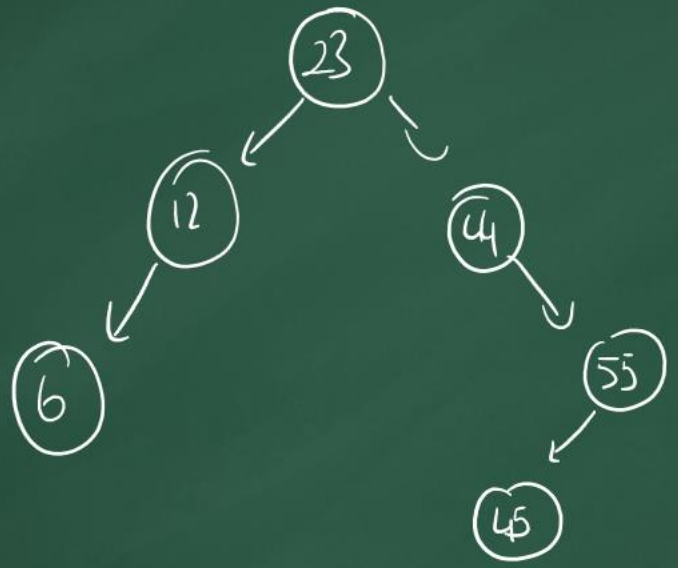
7 =

S → pop()

53

.... end while

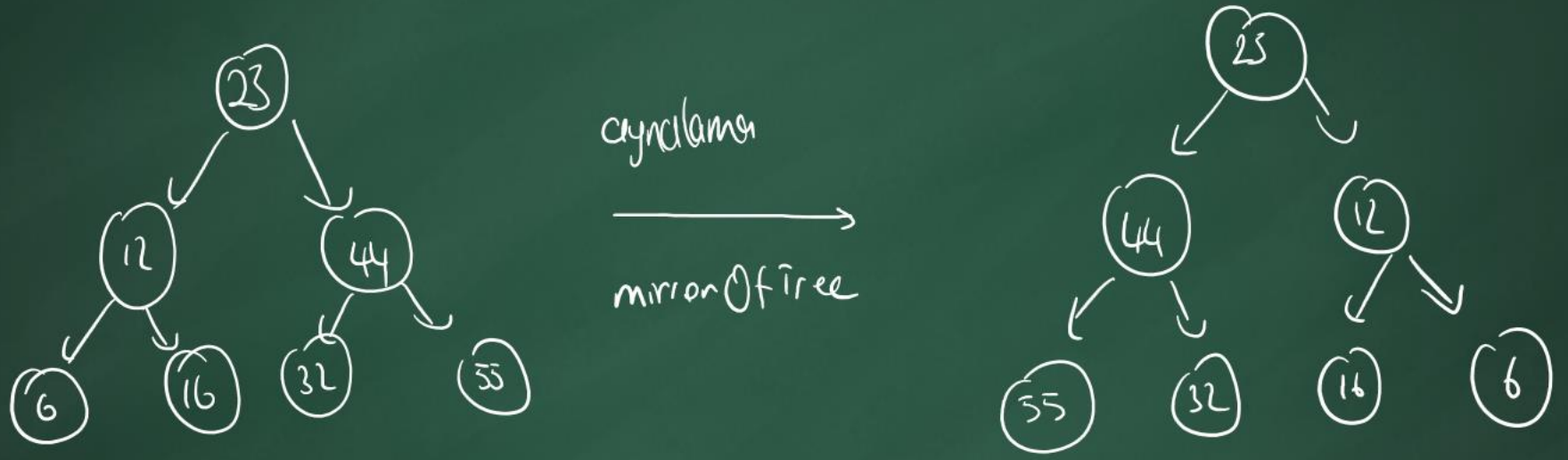
Problem: Kökten yapraklara olan yolların yazdırılması

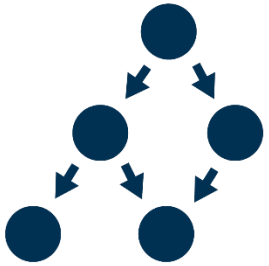


Çıktı → 23 - 12 - 6

23 - 44 - 55 - 45

Problem : Bir ağacın aynalanmasını / simetrisini sağlayan metot tasarımını gerçekleştiriniz.





Veri Yapıları ve Algoritmalar

ZAFER CÖMERT
Öğretim Üyesi

BTK AKADEMİ