

VERİ YAPILARILARI VE ALGORİTMALAR

Binary Search Tree

Giriş

1. İkili arama ağaçları (Binary Search Trees, BSTs)
 1. Ekleme
 2. Inorder, preorder, postorder dolaşma
 3. Minimum değerli elemanı bulma
 4. Maksimum değerli elemanı bulma
 5. Herhangi bir ifadeyi bulma
 6. Silme
 1. Yaprak silme
 2. Tek düğümü olan bir düğümü silme
 3. İki düğümü olan bir düğümü silme

Binary Search Tree (BST)

- İkili arama ağacı üzerindeki yönetimsel fonksiyonları gerçekleştirmek üzere **BST** sınıf tasarımını gerçekleştiriyoruz.
- Söz konusu sınıf ile ağaca ekleme yapma, ağaç üzerinde dolaşma, ağaç üzerindeki minimum ve maksimum değerleri bulma gibi yönetimsel işlevlerin tasarımı gerçekleştiriyoruz.
- Öncelikle BST içinde **kök düğüm** ve **yapılandırıcı metot tasarımı** gerçekleştiriyoruz.

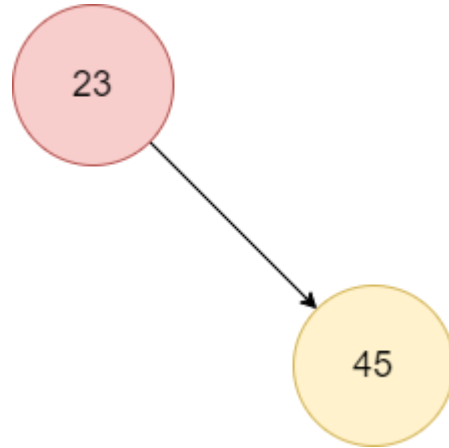
BST Ekleme

- BST ekleme yapmadan önce elemanın ekleneceği yerini bulmak gerekir.
- Uygun yeri bulmak için **Find** metodunda takip edilen adımların tekrar edilmesi gerekir.

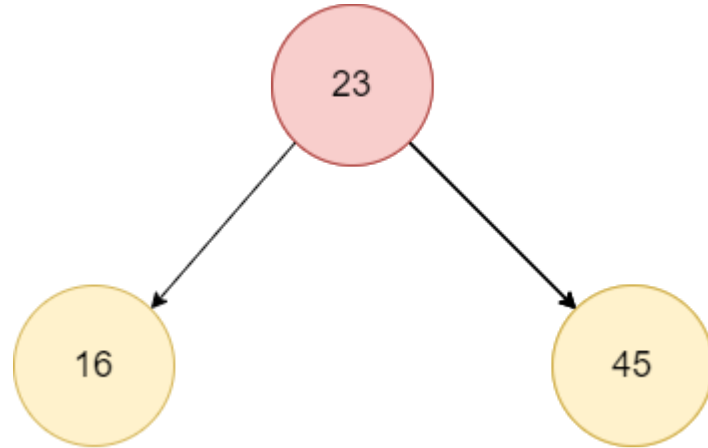
BST Ekleme



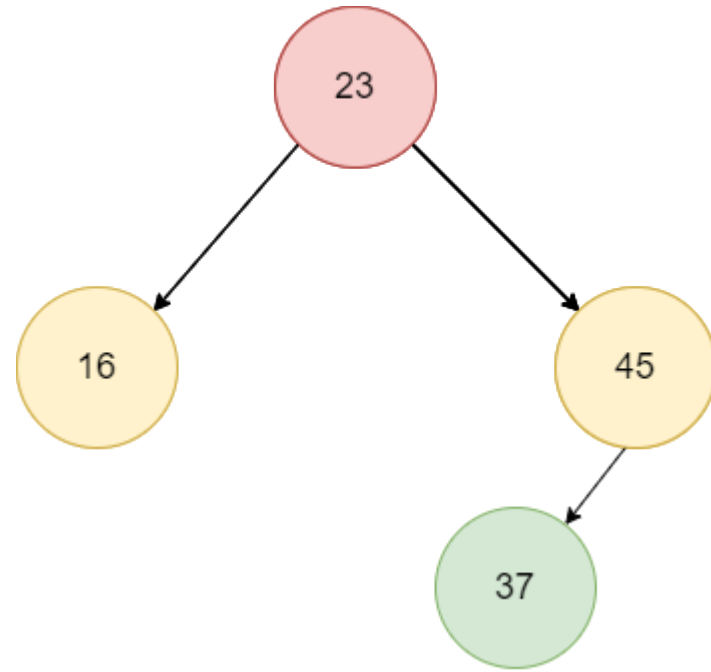
BST Ekleme



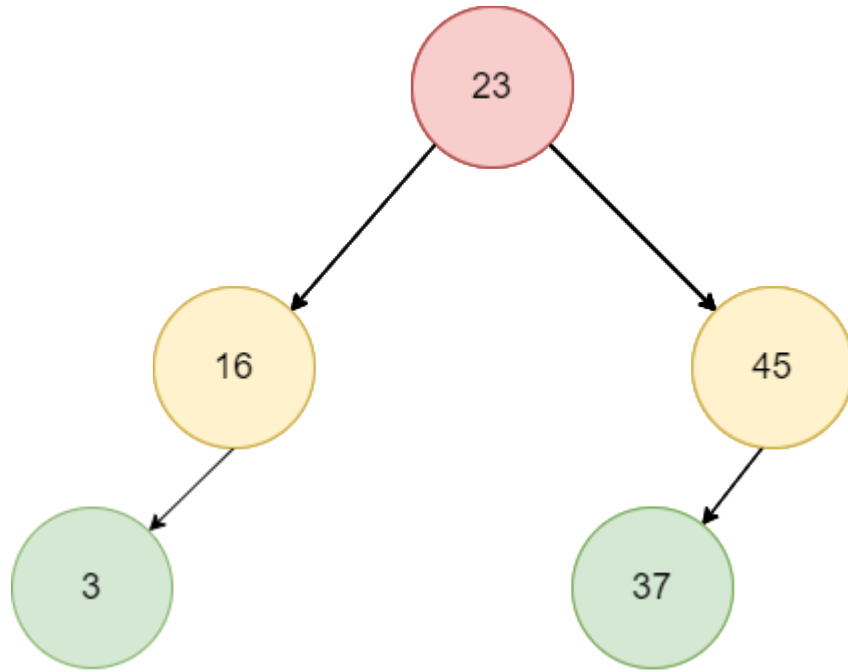
BST Ekleme



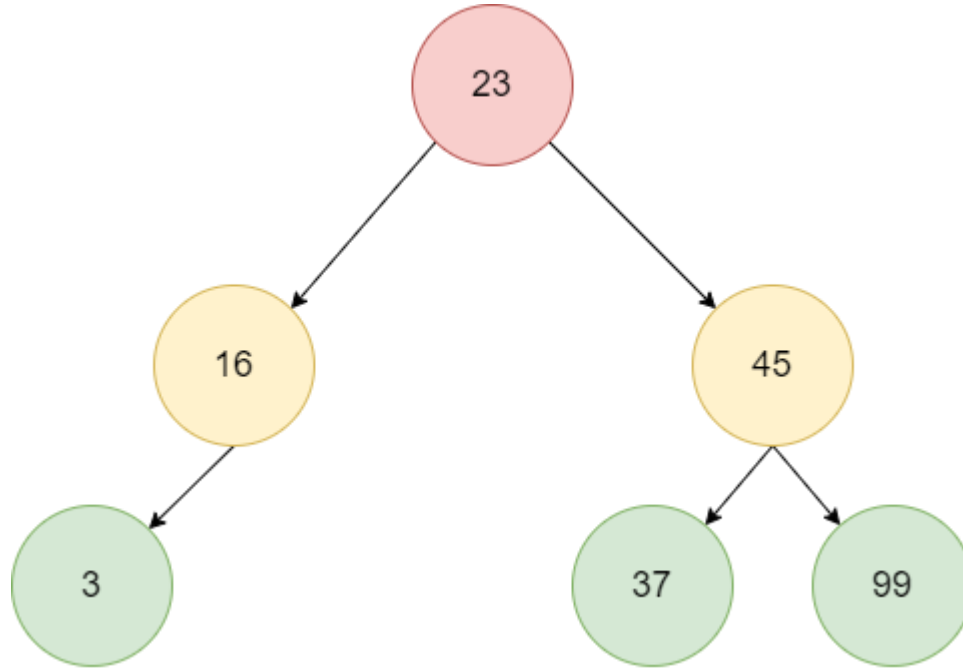
BST Ekleme



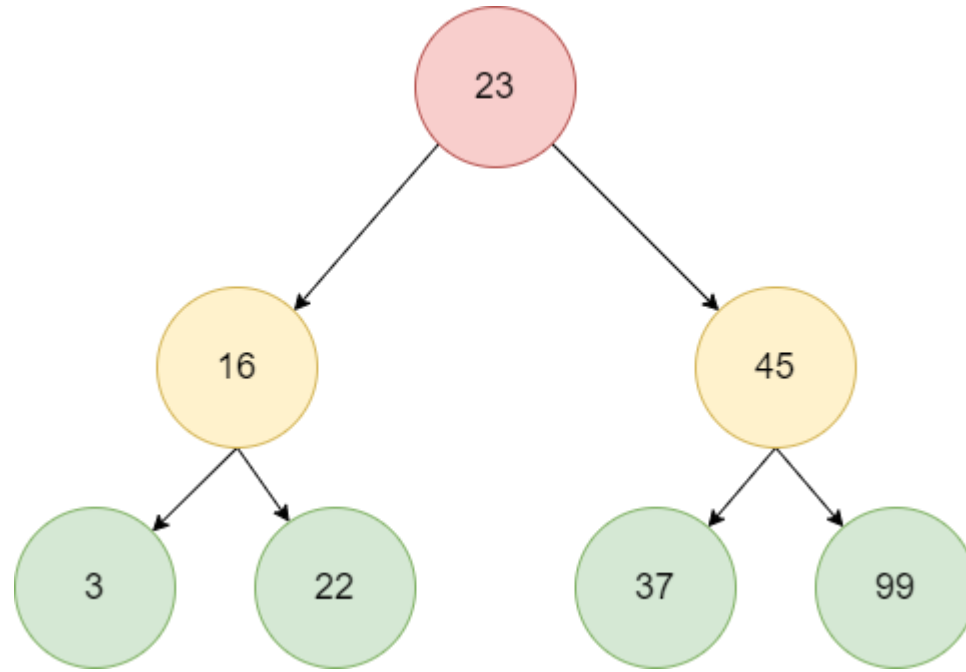
BST Ekleme



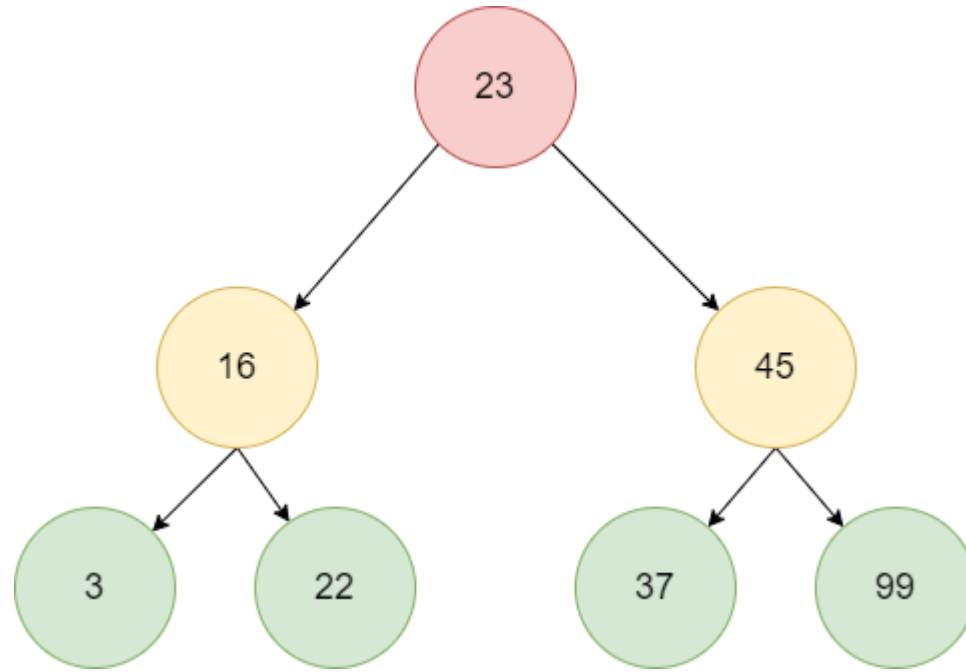
BST Ekleme



BST Ekleme



BST Ekleme



BST İşlevler

- **Ana işlevler**

- Arama / Minimum bulma / Maksimum bulma
- Ekleme
- Silme

- **Yardımcı işlevler**

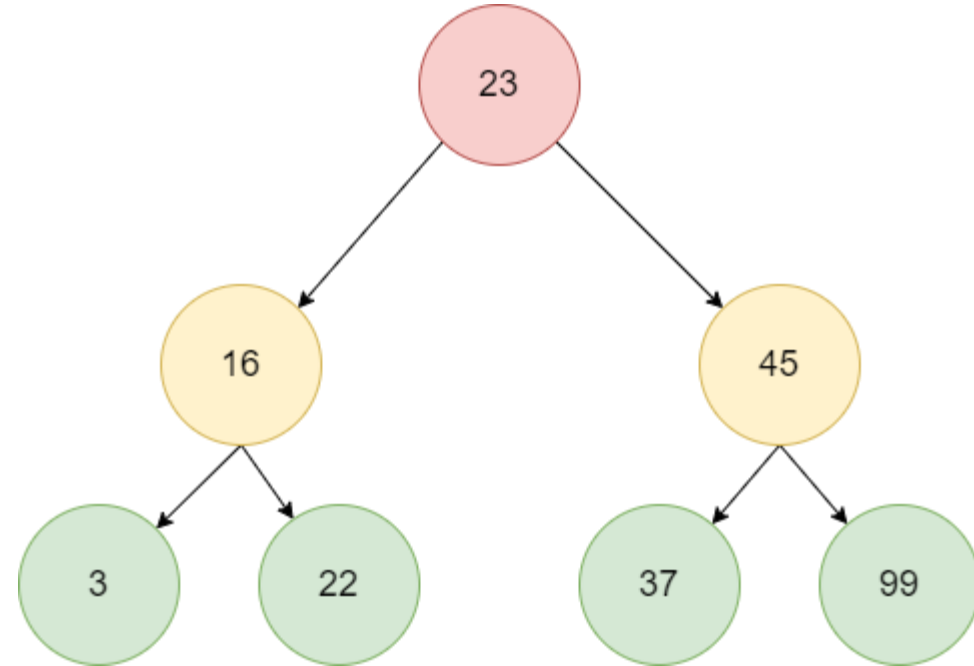
- Ağaç iki arama ağacı mı değil mi?
- İkili arama ağacında sıralama

BST ilişkin Notlar

- Kök düğümü ağacı her zaman sağ ve sol olarak iki parçaya böldüğünden ikili arama ağacı sıralı bir dizi oluşturur.
- Problemlerde öncelikle sol alt ağaç sonra kök ve daha sonra sağ alt ağaç dikkate alınır.

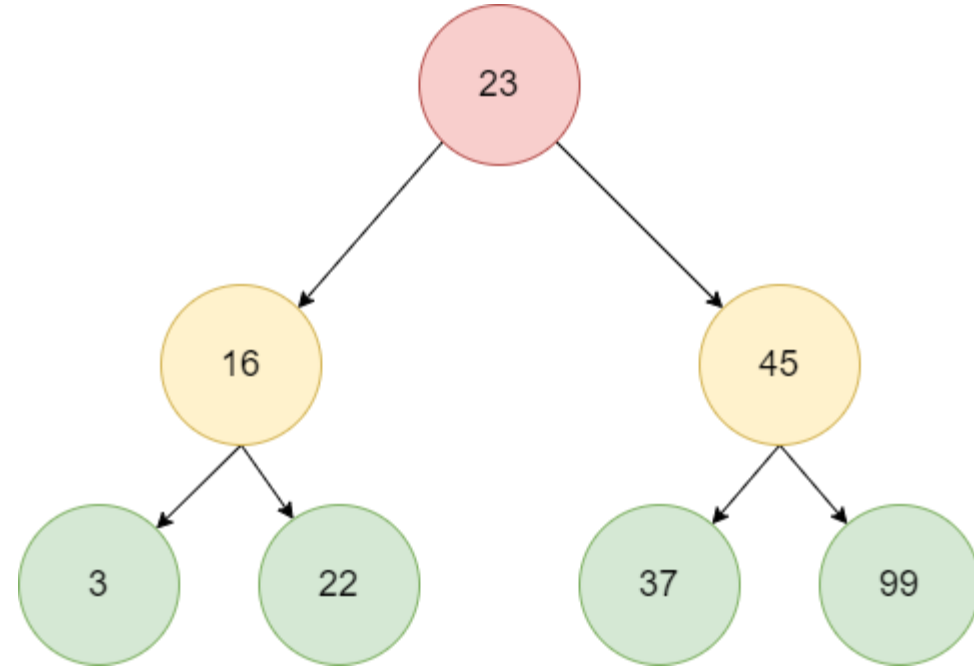
BST en küçük elemanın bulunması

- BST en küçük eleman en solda yer alan düğümdeki elemandır.
- Tüm elemanların sol düğümde toplandığı bir sol eğri ağaç yapısında bu işlemin maliyeti en kötü durumda $O(n)$ olur.

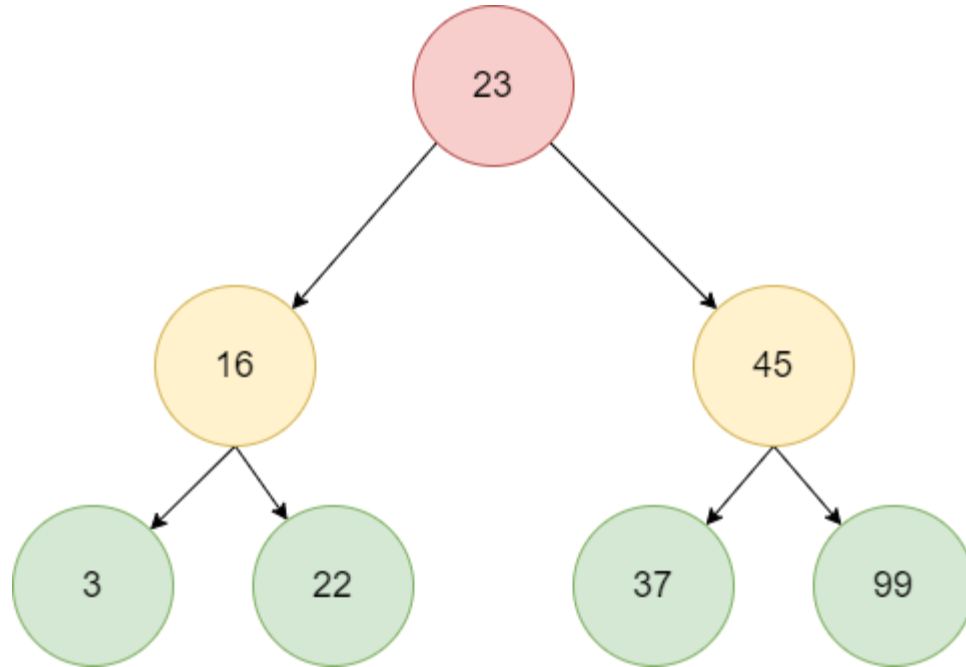


BST en büyük elemanın bulunması

- BST en büyük eleman en sağda yer alan düğümdeki elemandır.
- Tüm elemanların sağ düğümde toplandığı bir sağ eğri ağaç yapısında bu işlemin maliyeti en kötü durumda $O(n)$ olur.



Inorder Dolaşma



3

16

22

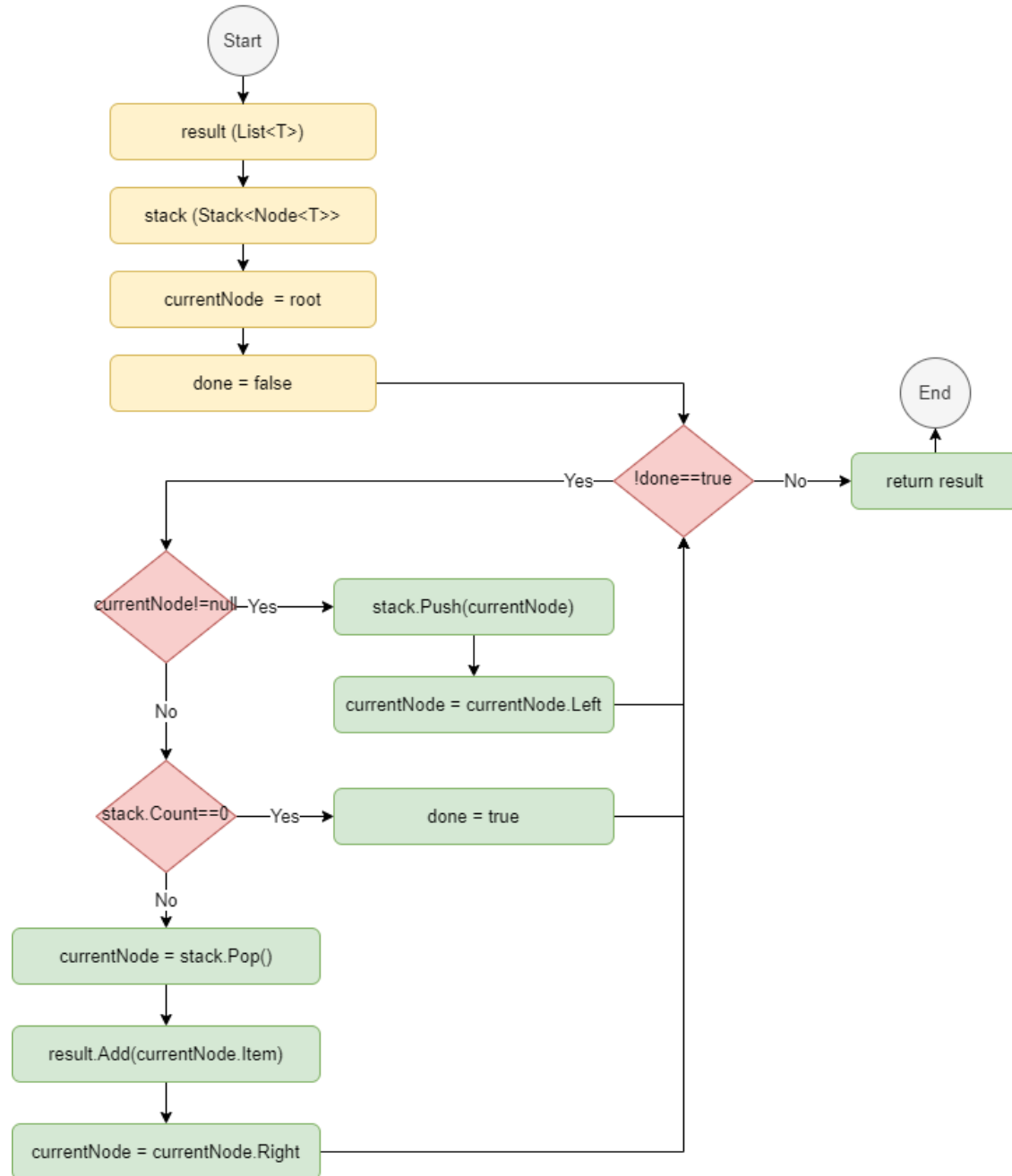
23

37

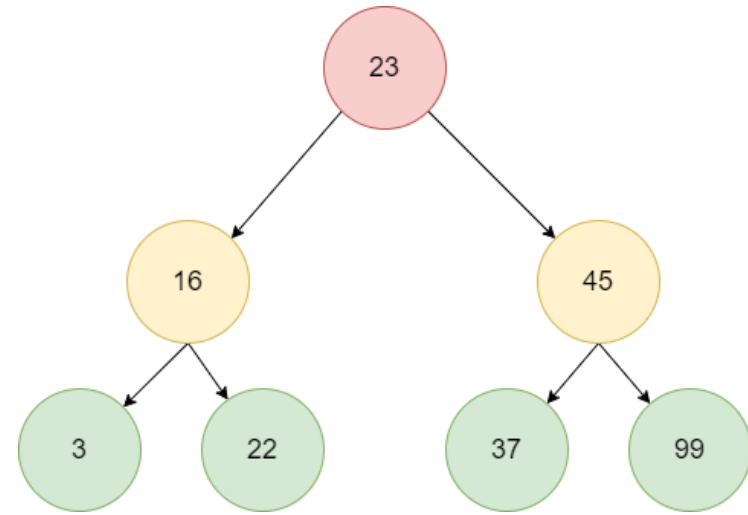
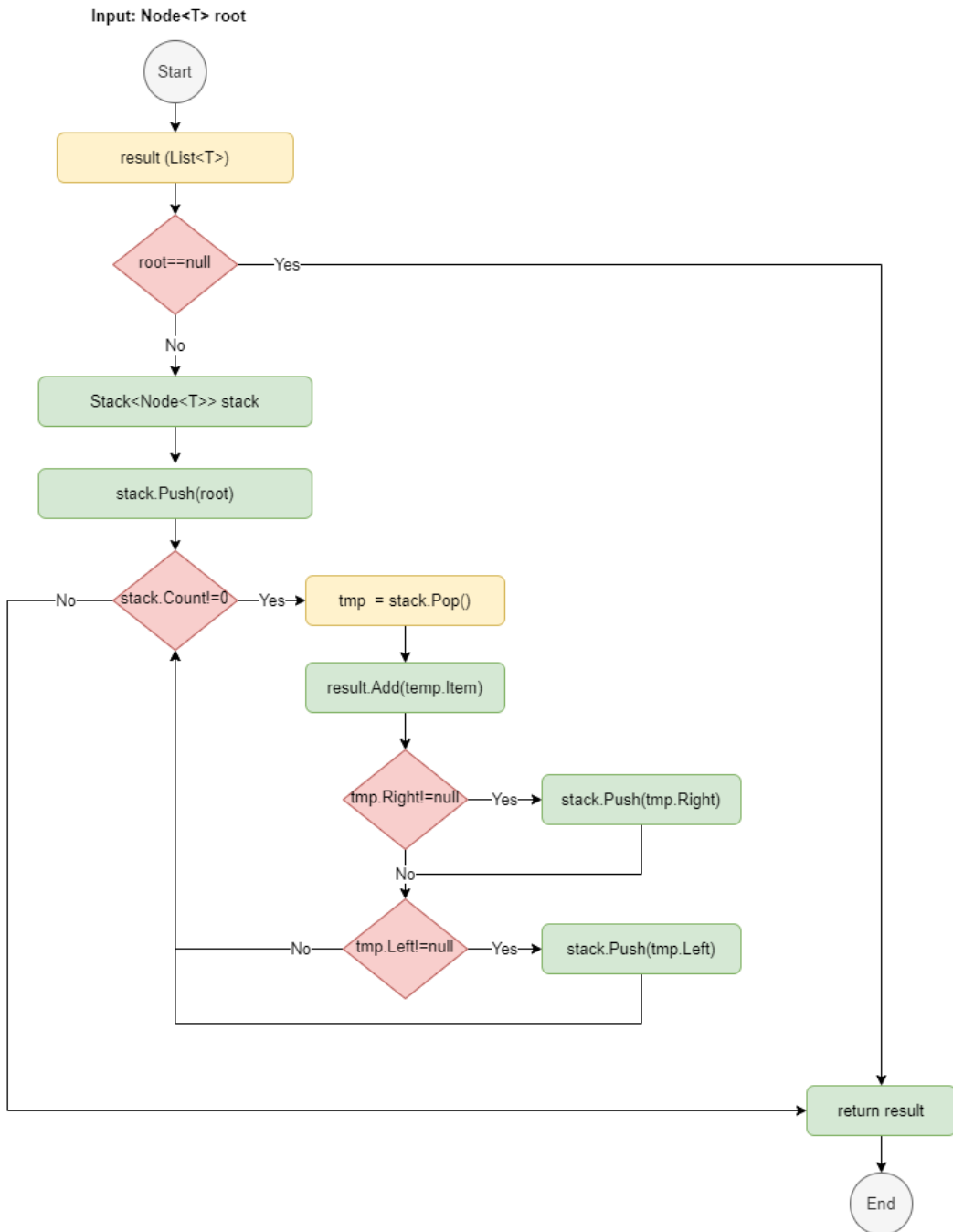
45

99

Input: Node<T> root

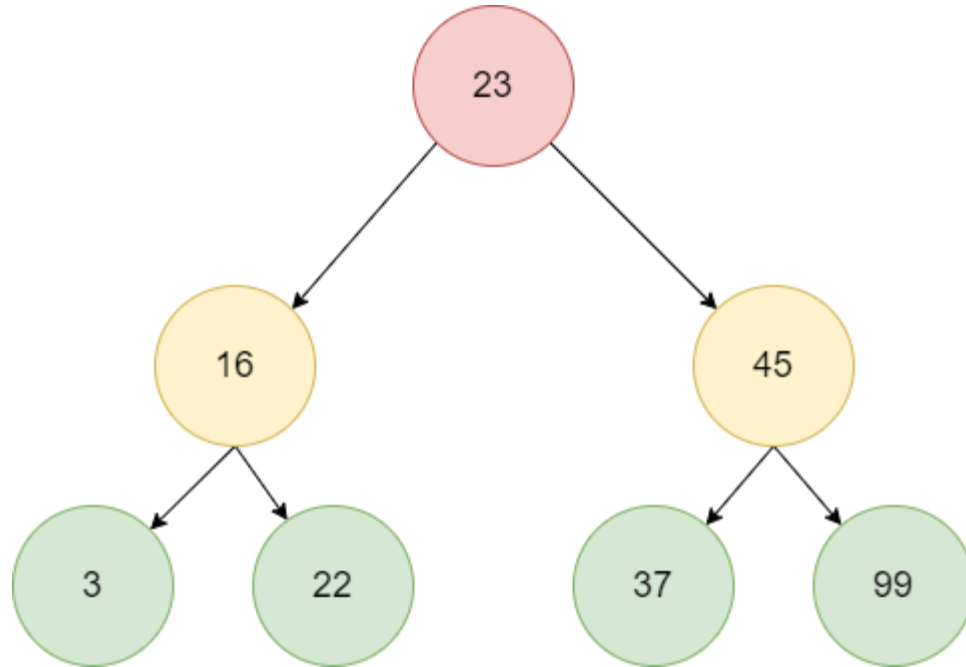


Non-recursive InOrder Traversal



Non-recursive PreOrder Traversal

Preorder Dolaşma



23

16

3

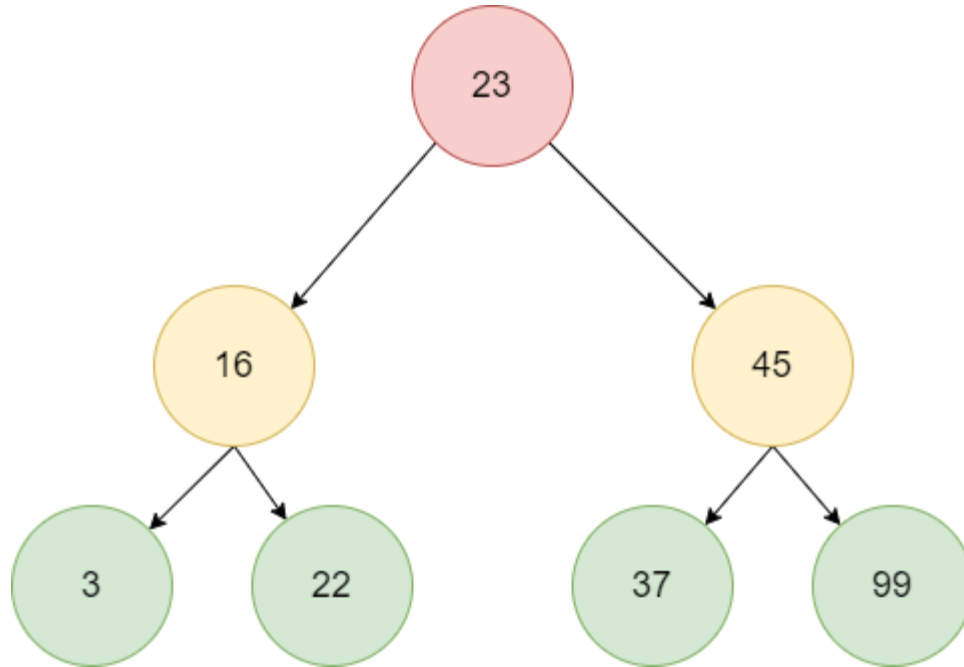
22

45

37

99

Postorder Dolaşma



3

22

16

37

99

45

23

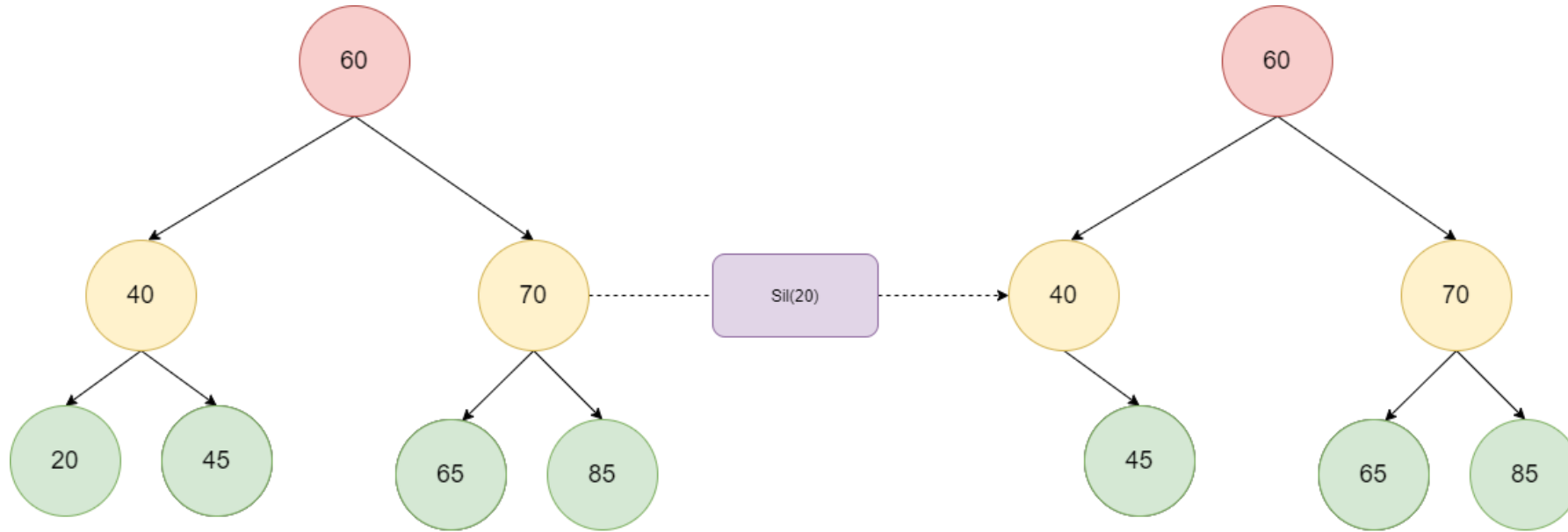
Ağaçtan eleman silme

Silinecek düğüm yaprak ise;

- Silinecek eleman bir yaprak ise, onun ailesine NULL değer döndürülür.
- Bunun anlamı çocuk işaretçi NULL olur.

Ağaçtan eleman silme

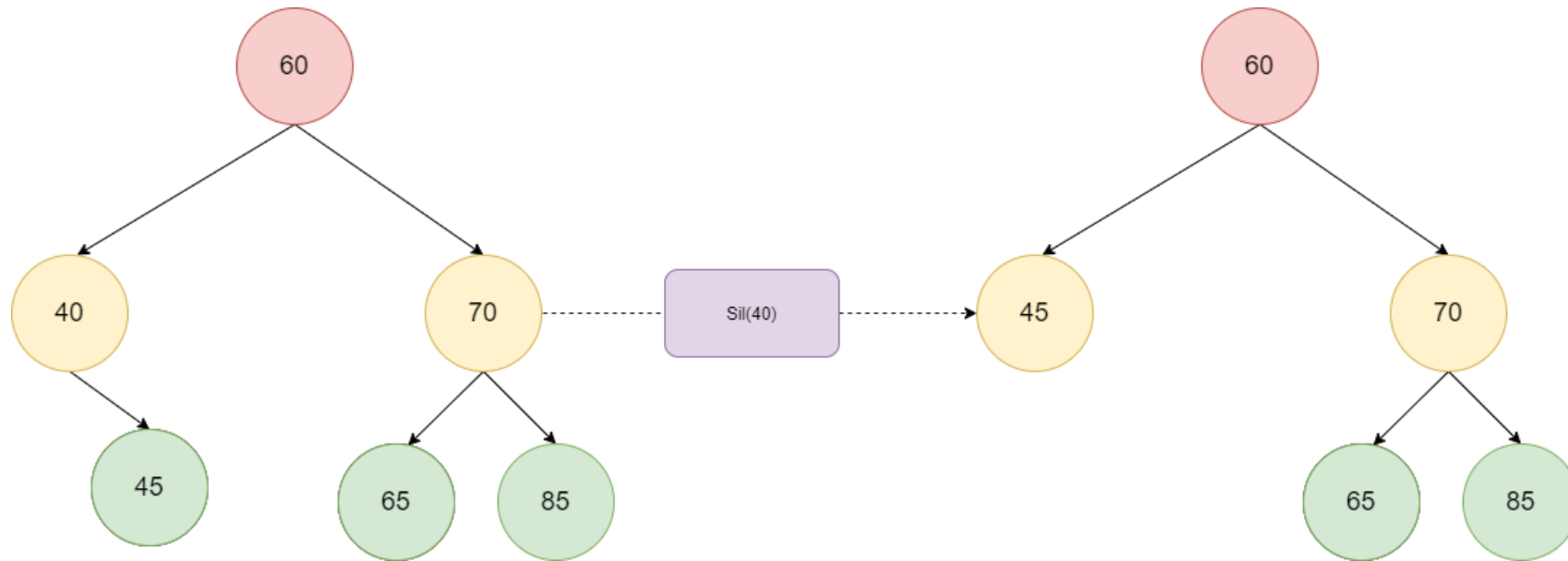
Silinecek düğüm yaprak ise;



Ağaçtan bir yaprağın silinmesi

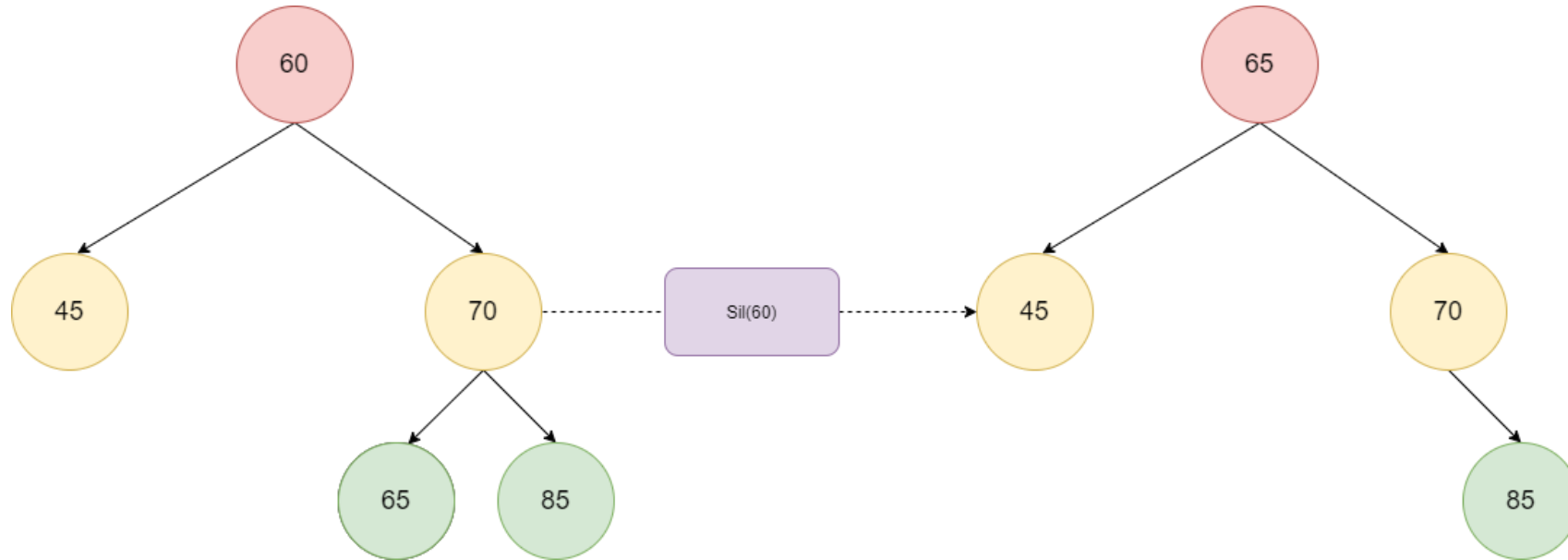
Ağaçtan eleman silme

Silinecek düğümün tek çocuğu var ise ;

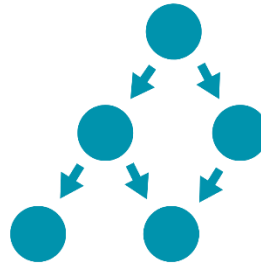


Silinecek düğümün bir alt ögesi vardır.

Ağaçtan eleman silme



Silinecek düğümün iki çocuğu vardır.



Veri Yapıları ve Algoritmalar

ZAFER CÖMERT

Öğretim Üyesi