

FÖRELÄSNING 8

Datastrukturer och algoritmer
KYH – 2022 HT

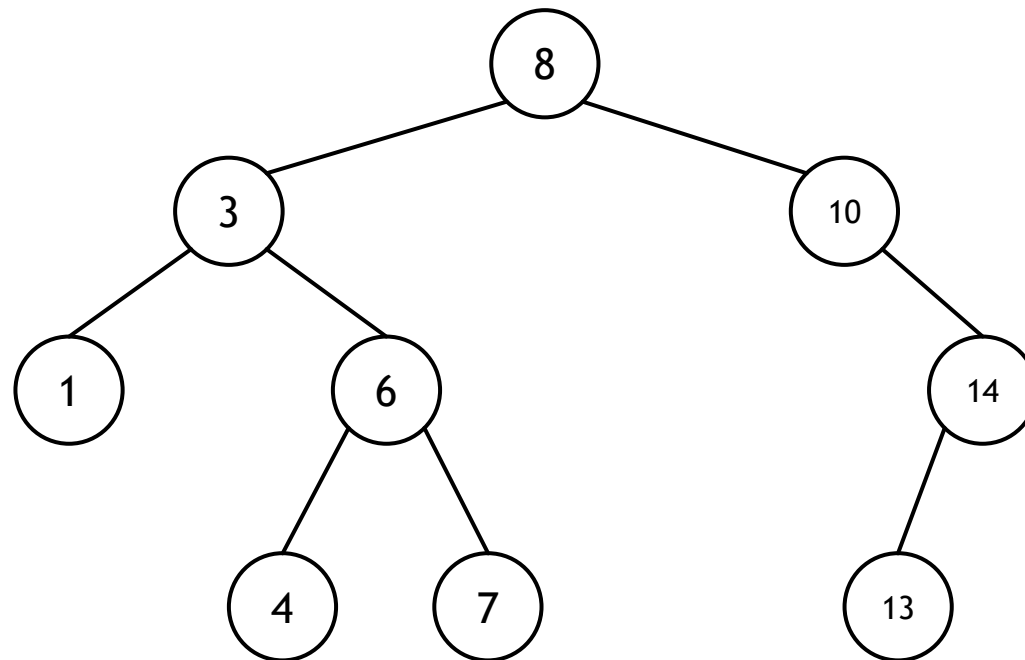
Andreas Nilsson Ström

Agenda

- ▶ Binärt sökträd
- ▶ Repetition
- ▶ Exempeltenta

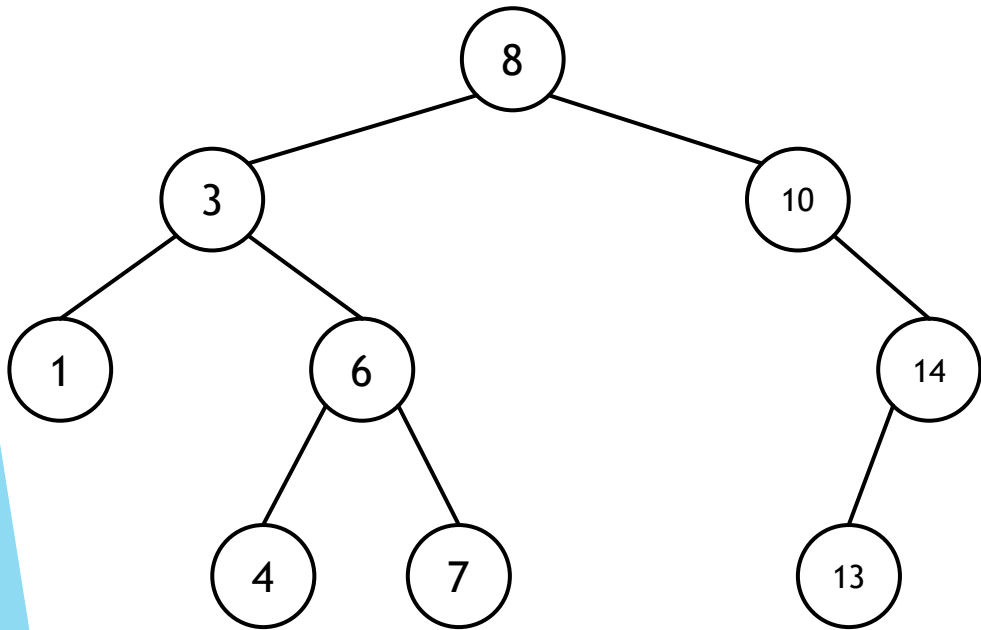
Binärt sökträd (BST)

- ▶ (A.K.A. Sorterat binärträd)
- ▶ Ett binärt träd som är garanterat att alltid vara sorterat
- ▶ Givet en nod så är alla mindre värden till vänster, alla större till höger



Sökning

- Kan göras rekursivt eller iterativt, med början på rot-noden

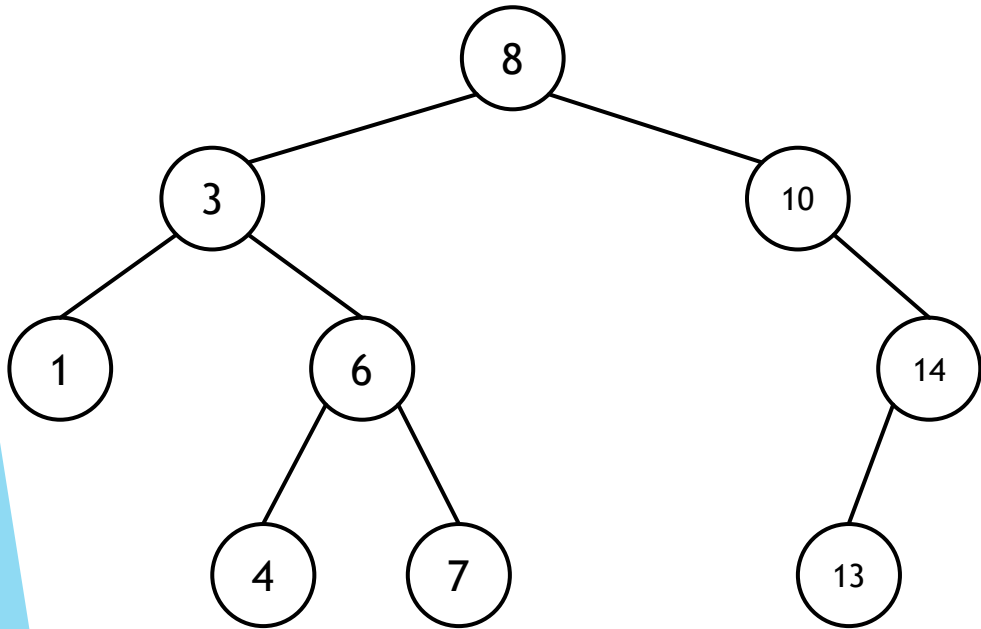


```
Recursive-Search(node, value)
  if node == NIL or value == node.value
    return node

  if value < node.value
    return Recursive-Search(node.left, value)
  else
    return Recursive-Search(node.right, value)
  end if
```

Sökning

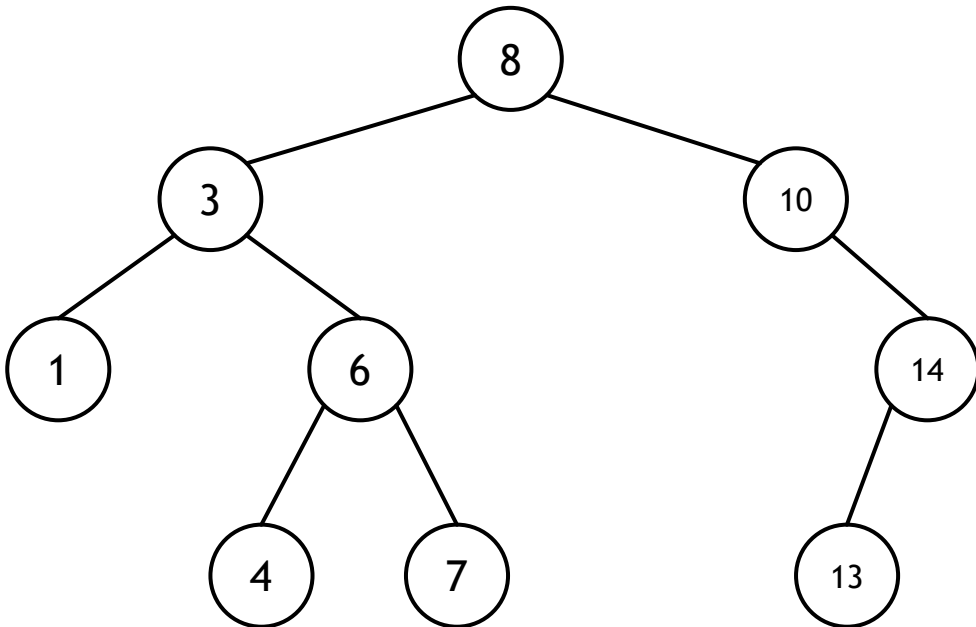
- Kan göras rekursivt eller iterativt, med början på rot-noden



```
Iterative-Search(node, value)
  while node ≠ NIL and value ≠ node.value
    if value < node.value then
      node := node.left
    else
      node := node.right
    end if
  repeat
  return node
```

Traversering: Utskrift av alla värden

- ▶ Kan göras rekursivt eller iterativt, med början på rot-noden
- ▶ Förra gången: Pre-order och Post-order traversal
- ▶ Utskrift är Mid-order traversal



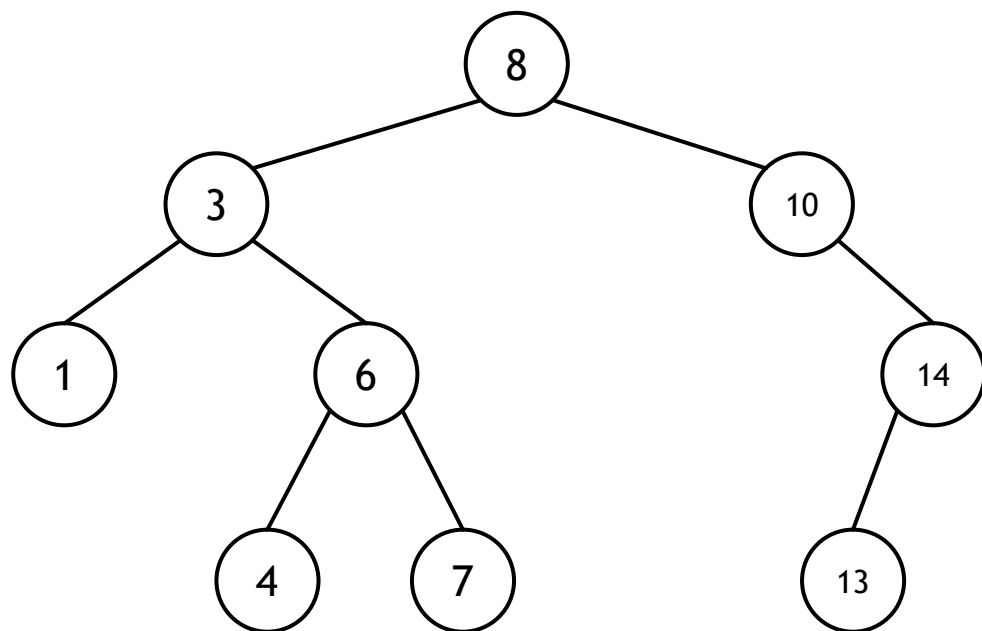
```
List(node)
  if node.left ≠ NIL
    List(node.left)

  print node.value

  if node.right ≠ NIL
    List(node.right)
```

Infoga nytt värde

- ▶ Traversera vänster/höger genom noderna
- ▶ Om vi behöver gå till en nod som är None, infoga värdet där



```
Insert(node, value):
```

```
    if node == NIL:  
        return Node(value)
```

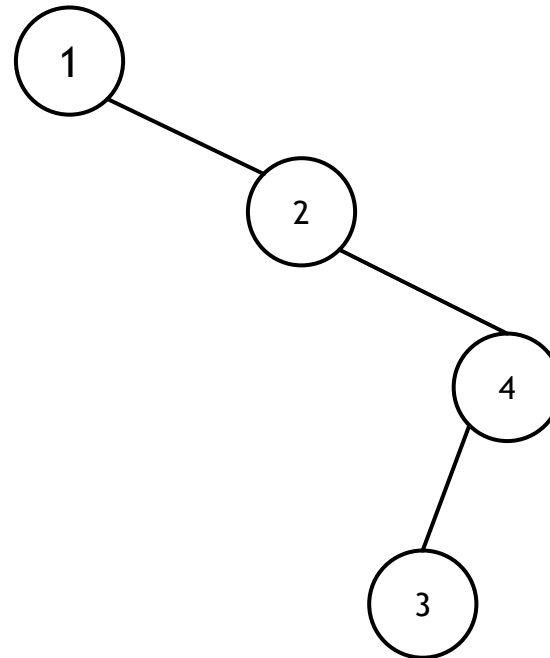
```
    if value < node.data:  
        node.left = Insert(node.left, value)
```

```
    else:  
        node.right = Insert(node.right, value)
```

```
    return node
```

Operationer

- ▶ Sökning
 - ▶ Worst case: $O(N)$
 - ▶ Average: $O(\log N)$
- ▶ Insert
 - ▶ Worst case: $O(N)$
 - ▶ Average: $O(\log N)$
- ▶ Delete
 - ▶ Worst case: $O(N)$
 - ▶ Average: $O(\log N)$



Mer läsning

Bok

- ▶ **Problem Solving with Algorithms and Data Structures Using Python**
- ▶ [Länk](#)

Dagens material: 6.11 - 6.14

