

ALGORITHMS & DATA STRUCTURES

Depth-First Traversal

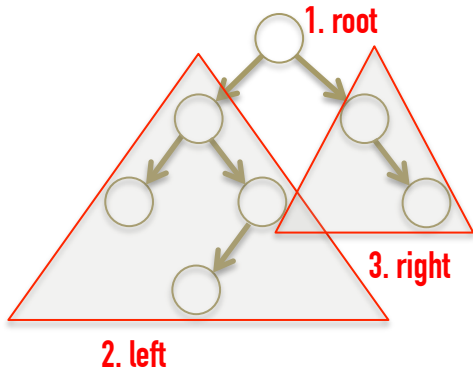
DEPARTMENT OF COMPUTING

A. Beghelli
Exam revision – May 2019

Depth-First Traversal

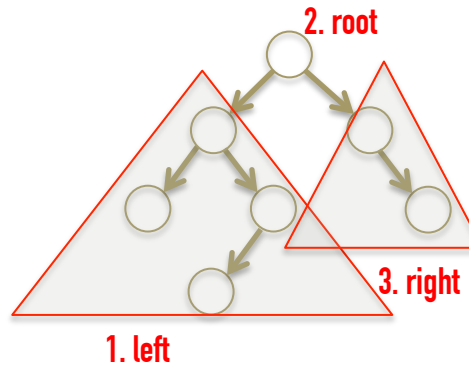
→ root
left
right

Pre-Order



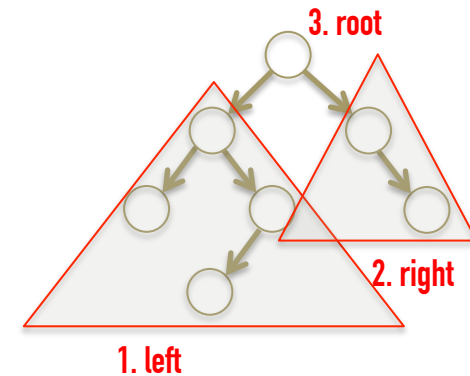
left
→ root
right

In-Order



left
right
→ root

Post-Order



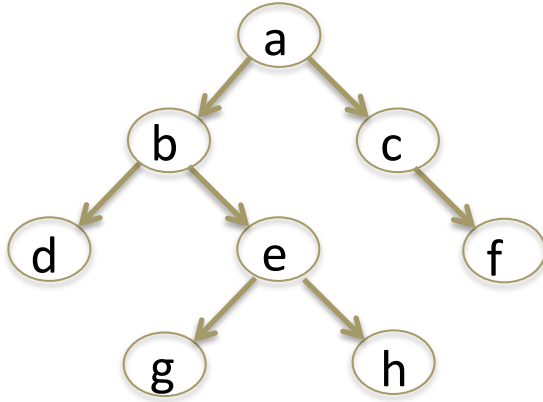
```
function pre-order(T)
  if ¬null?(T) then
    → visit(T)
    pre-order(left(T))
    pre-order(right(T))
  end if
end function
```

```
function in-order(T)
  if ¬null?(T) then
    in-order(left(T))
    → visit(T)
    in-order(right(T))
  end if
end function
```

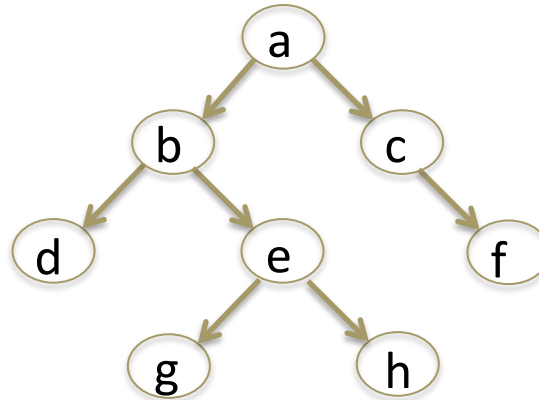
```
function post-order(T)
  if ¬null?(T) then
    post-order(left(T))
    post-order(right(T))
    → visit(T)
  end if
end function
```

Depth-First Traversal

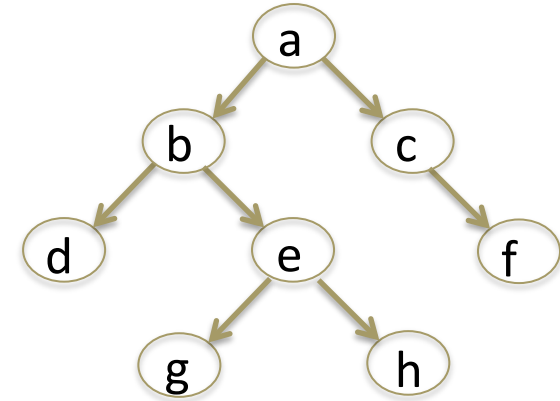
→ root
left
right **Pre-Order**



left
→ root
right **In-Order**



left
right
→ root **Post-Order**

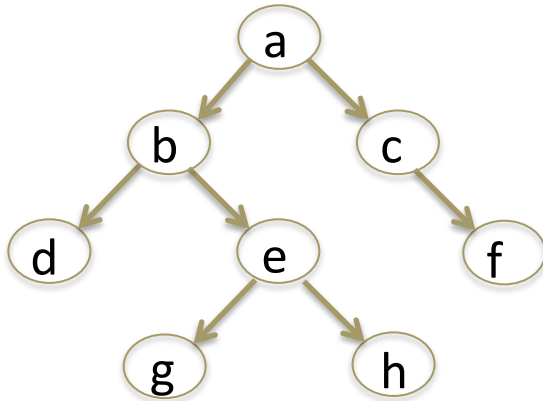


```
function pre-order(T)
  if ¬null?(T) then
    → visit(T)
    pre-order(left(T))
    pre-order(right(T))
  end if
end function
```

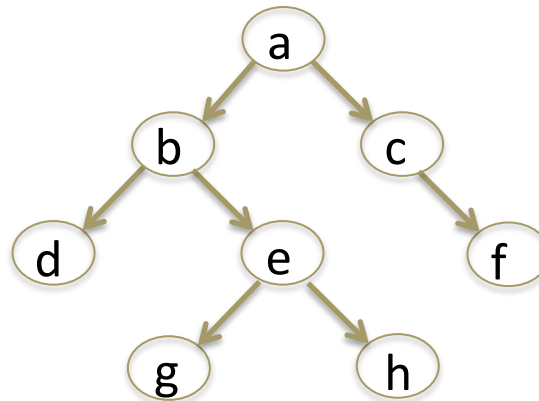
Left sub-tree Right sub-tree
[a,b,d,e,g,h,c,f]
root precedes everything

Depth-First Traversal

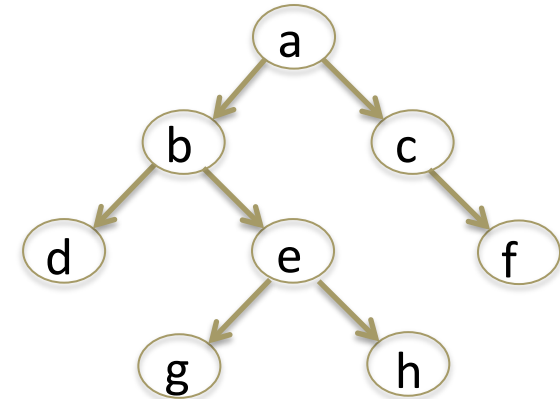
→ root
left
right **Pre-Order**



left
→ root
right **In-Order**



left
right
→ root **Post-Order**



```
function pre-order(T)
  if ¬null?(T) then
    → visit(T)
    pre-order(left(T))
    pre-order(right(T))
  end if
end function
```

Left sub-tree Right sub-tree
[a,b,d,e,g,h,c,f]
root precedes everything

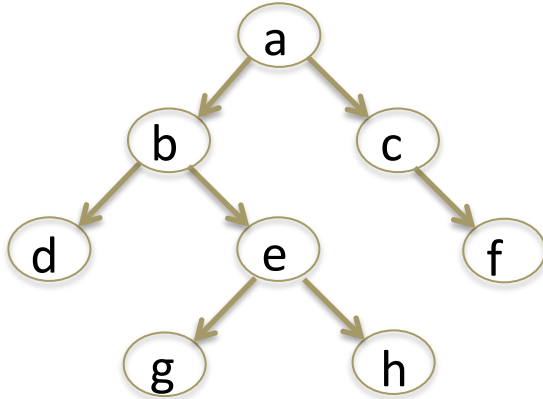
```
function in-order(T)
  if ¬null?(T) then
    in-order(left(T))
    → visit(T)
    in-order(right(T))
  end if
end function
```

Left sub-tree Right sub-tree
[d,b,g,e,h,a,c,f]
root in the middle

Depth-First Traversal

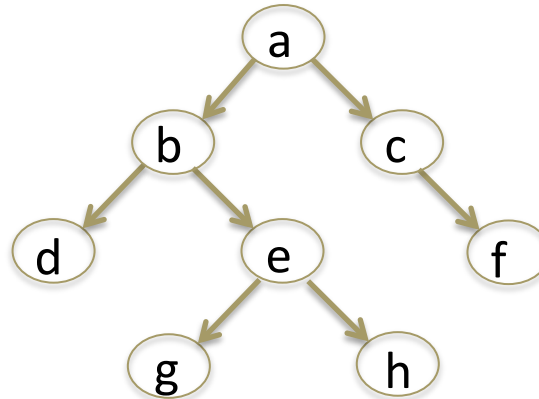
→ root
left
right

Pre-Order



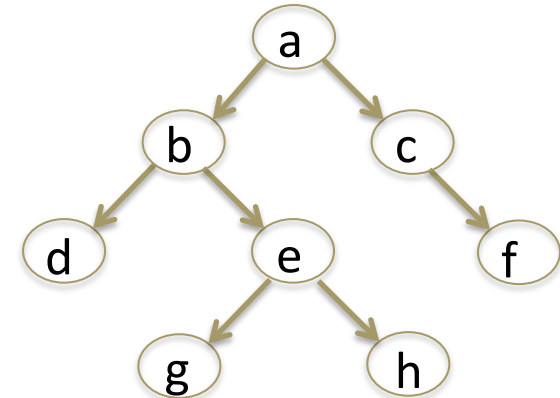
left
→ root
right

In-Order



left
right
→ root

Post-Order



```
function pre-order(T)
  if ¬null?(T) then
    → visit(T)
    pre-order(left(T))
    pre-order(right(T))
  end if
end function
```

Left sub-tree Right sub-tree
[a,b,d,e,g,h,c,f]
root precedes everything

```
function in-order(T)
  if ¬null?(T) then
    in-order(left(T))
    → visit(T)
    in-order(right(T))
  end if
end function
```

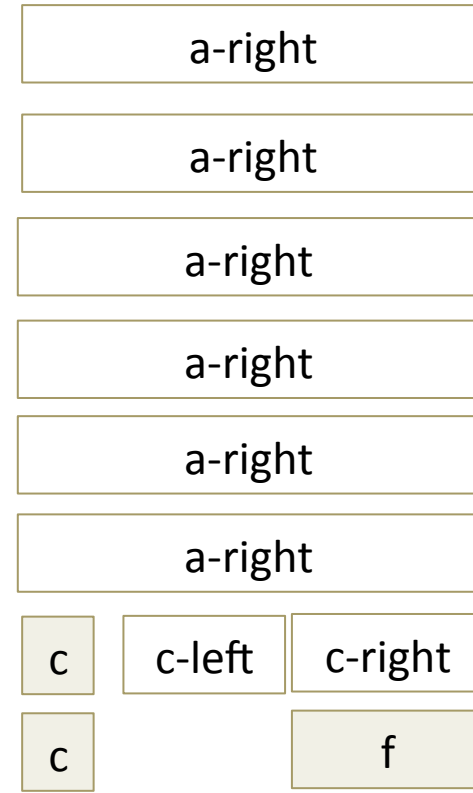
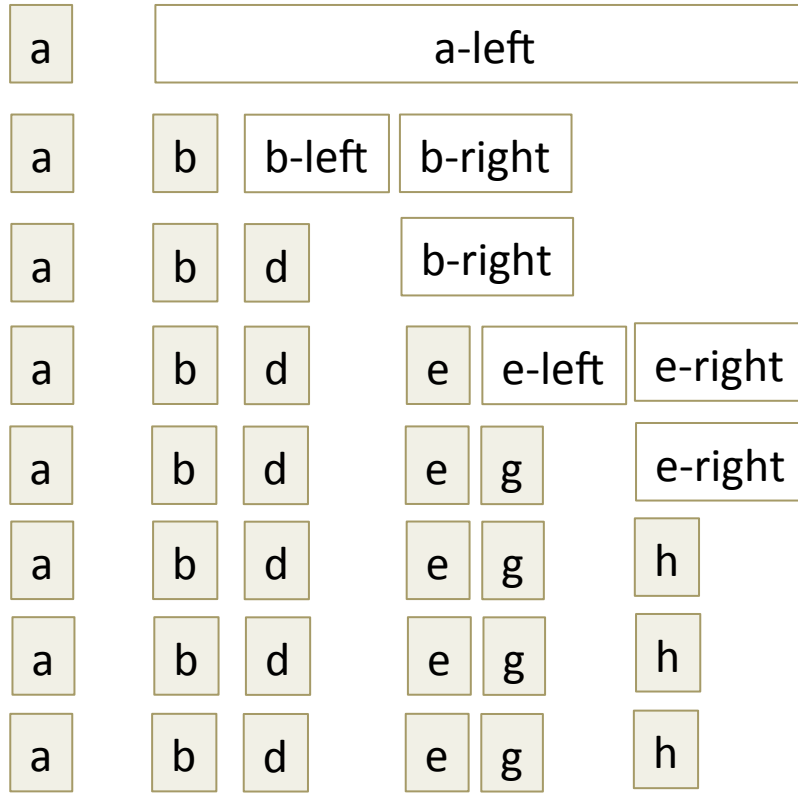
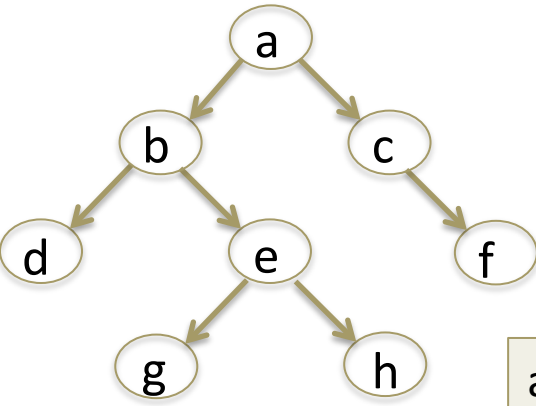
Left sub-tree Right sub-tree
[d,b,g,e,h,a,c,f]
root in the middle

```
function post-order(T)
  if ¬null?(T) then
    post-order(left(T))
    post-order(right(T))
    → visit(T)
  end if
end function
```

Left sub-tree Right sub-tree
[d,g,h,e,b,f,c,a]
root after everything

Pre-order (root **pre**cedes everything)

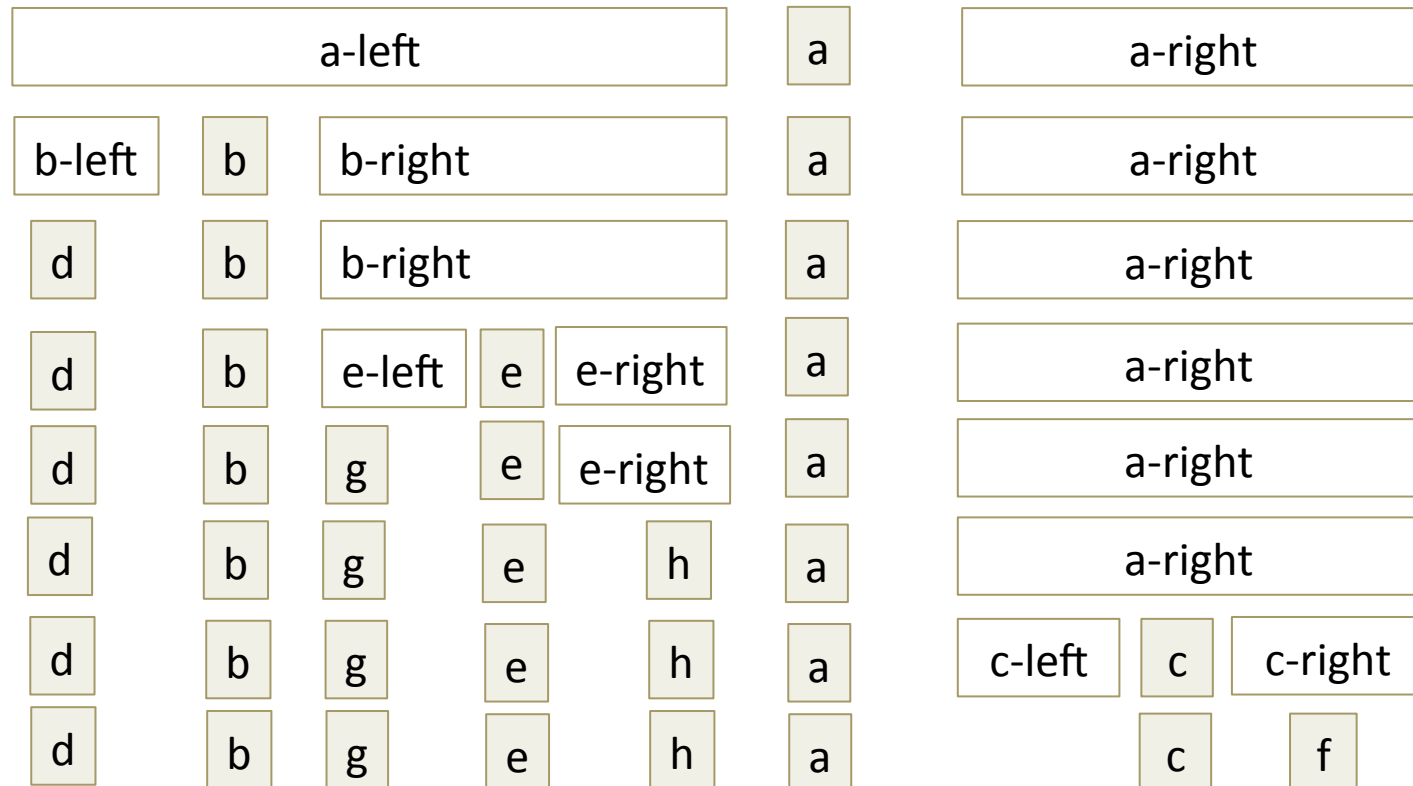
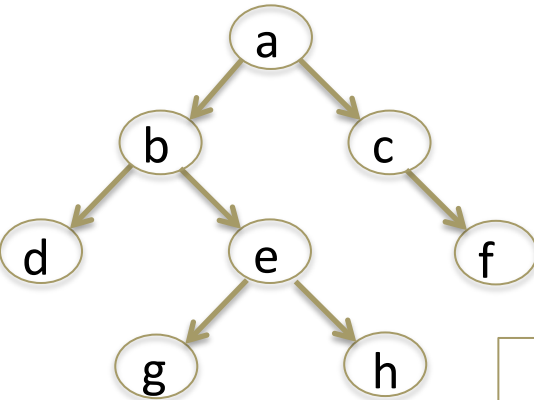
→ root
left
right **Pre-Order**



In-order (root **in** the middle)

left
→ **root**
right

In-Order



Post-order (root **post**erior to everything)

left
right
→ root

Post-Order

