KEY POINTS

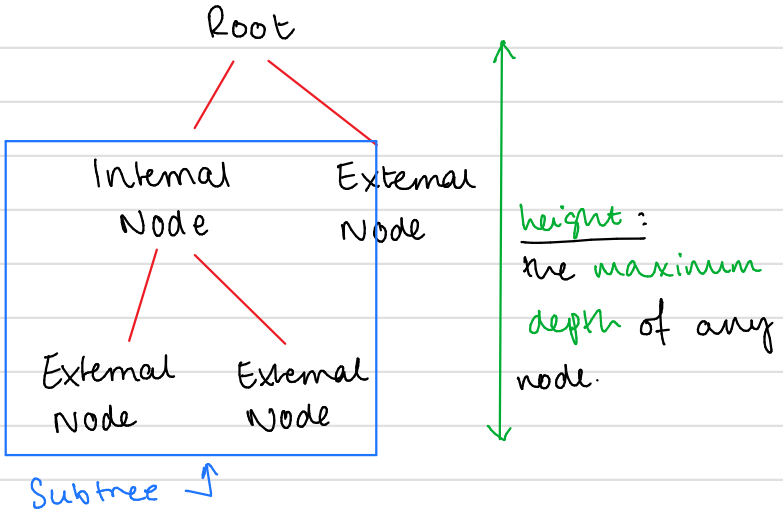
Trees - an abstract model of a hierarchical structure.

- A tree will have generic, accessor, query methods.
- One can use preorder and portorder traversal.

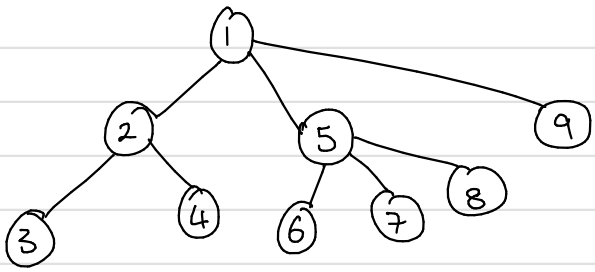
NAME/DATE/SUBJECT

Trees Introduction.

NOTES



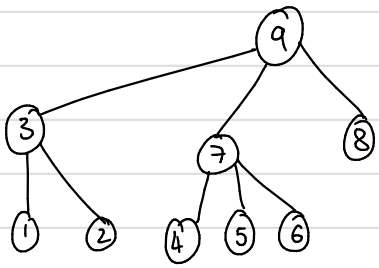
Portorder traversal



Print the node before children.

Preorder Traversal:

Print the node after its children.



SUMMARY

KEY POINTS

- Each node has at most two children.
  - ⇒ A proper binary tree will have exactly 2 children per node.
  - E.g. arithmetic or decision trees.

ADT Functions

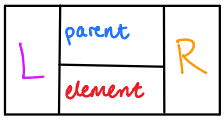
All return position

left(p) right(p) sibling(p) — null if failure

Traversal

- Inorder traversal means that a node is printed after the left subtree and before the right subtree.
- Useful for printing arithmetic expressions - use postorder trav. to evaluate the tree.

Linked Structure:



NAME/DATE/SUBJECT

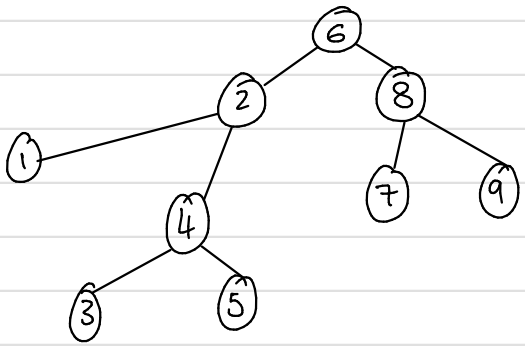
Binary Trees

NOTES

Properties of proper binary trees:

n	# nodes	$e = i + 1$	$e \leq 2^h$
e	# external nodes	$n = 2e - 1$	$h \geq \log_2 e$
i	# internal nodes	$h \leq i$	$h \geq \log_2 (n+1) - 1$
h	height	$h \leq \frac{(n-1)}{2}$	

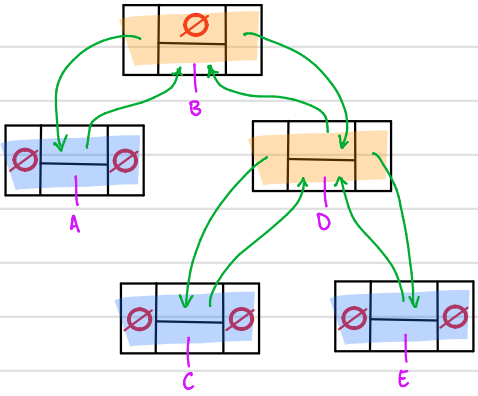
Inorder traversal :

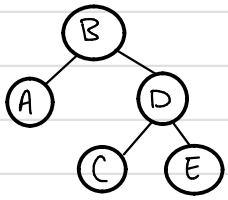


Algo inorder :

```
if left(v) ≠ null
    inorder(left(v))
print(v)
if right(v) ≠ null
    inorder(right(v))
```

Linked Structure





SUMMARY

KEY POINTS

You can store binary trees in an array by having f(p) store the node p:

f(0) = root

p is the left child of q:
f(p) = 2 · f(q) + 1

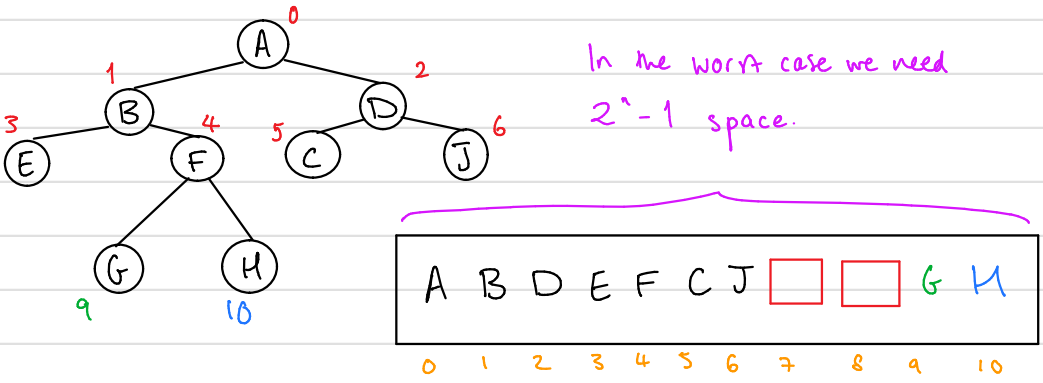
p is the right child of q:
f(p) = 2 · f(q) + 2

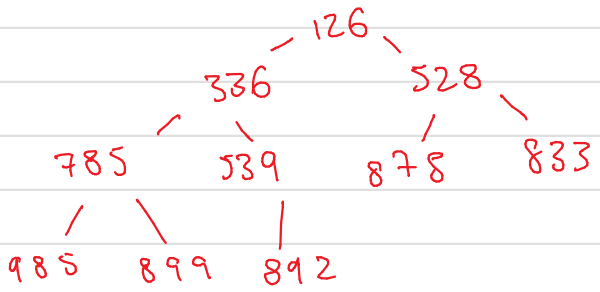
NAME/DATE/SUBJECT

Array based tree

NOTES

Binary tree in an array:





SUMMARY