Tree traversals

Today's announcements:

- MT1 Feb 4, 7-9:00p WOOD 2
- HW2 out, due Feb 5, 11:59p

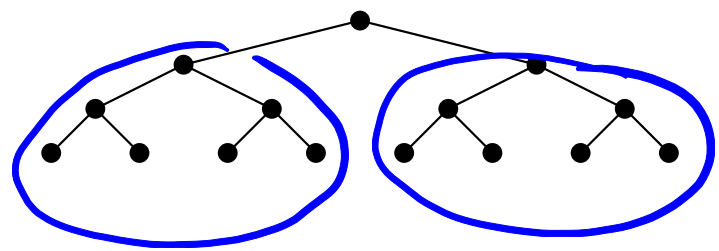
Today's Plan

- Trees and their traversals

Warm up: What is the number of nodes $N(h)$ in a perfect binary tree of height h?

$$N(h) = \sum_{i=0}^{h} 2^i$$

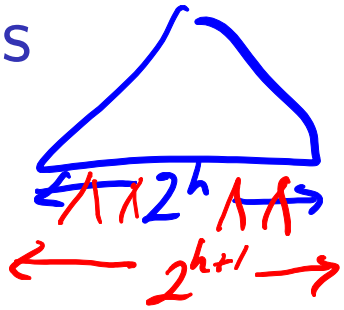
$$= 2^{h+1} - 1$$

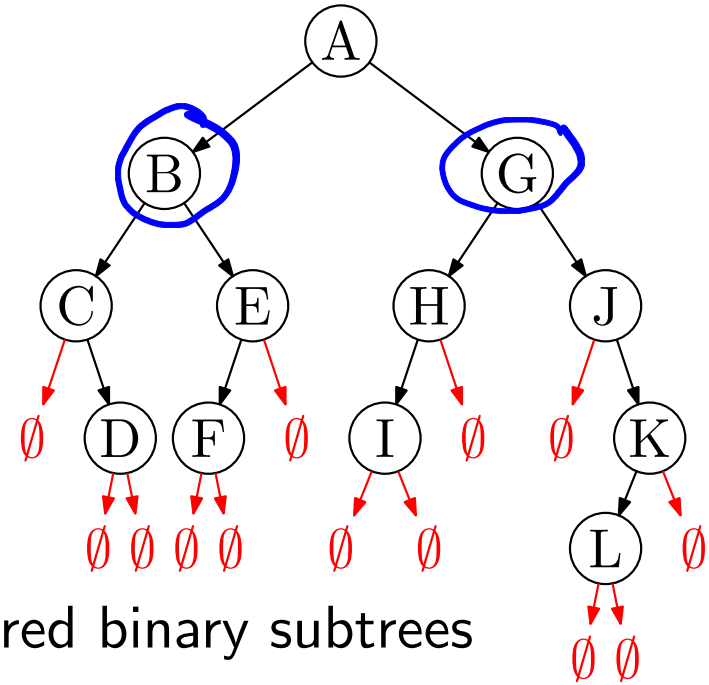


$$N(h) = 2N(h-1) + 1$$

$$N(0) = 1$$

Ordered Binary Trees





An ordered binary tree is

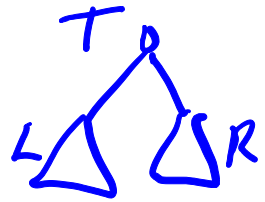
- empty ($\emptyset$), or
- root node with left and right ordered binary subtrees

Given height h binary tree,

- max # nodes = $2^{h+1} - 1$
- max # leaves = 2^h
- max # empty slots = 2^{h+1}

#emptys = #nodes + 1

Always ??

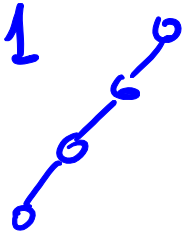


2 child root $e(T) = e(L) + e(R)$

by I.H. $= n(L) + 1 + n(R) + 1$

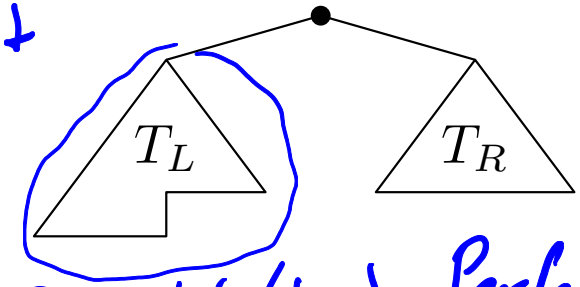
$= n(T) + 1$

min # nodes = $h + 1$

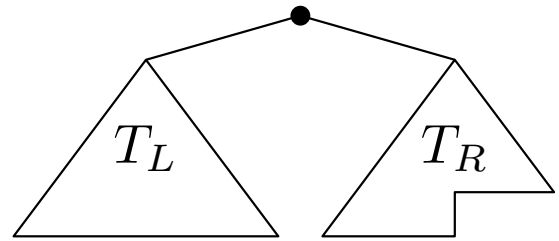


Complete binary tree

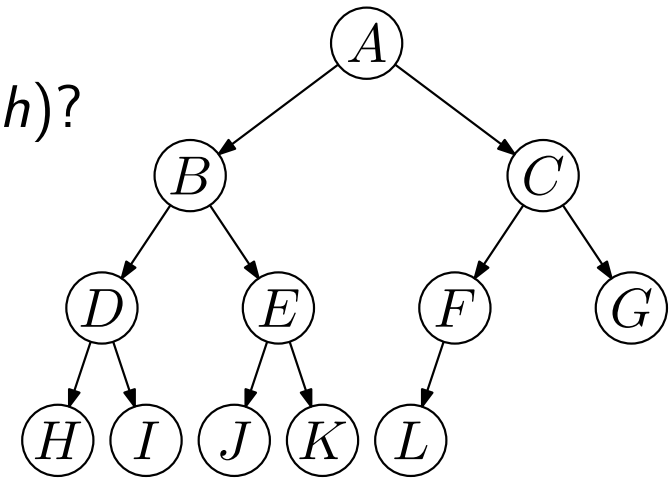
A complete tree of height h is



or



Min/Max number of nodes in Complete(h)?



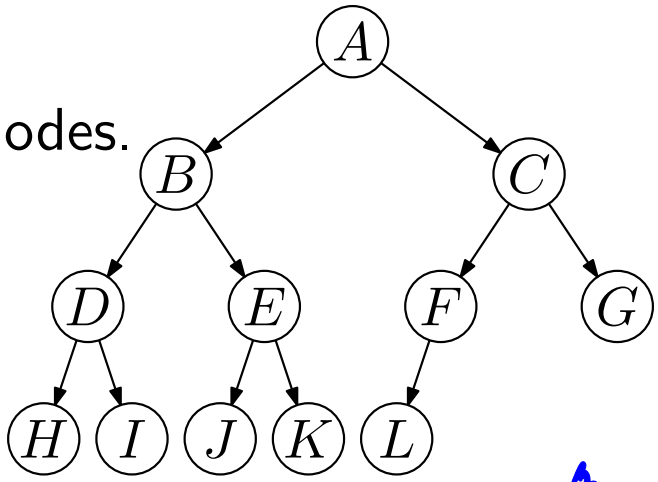
Complete binary tree height

Height of the complete binary tree with n nodes.

$H(1) = 0$

$H(2) = \cdots = H(3) = 1$

$H(4) = \cdots = H(7) = 2$



$H(2^h) = \cdots = H(2^{h+1} - 1) = h$

(2^h: n nodes; $2^{h+1} - 1$: n nodes)

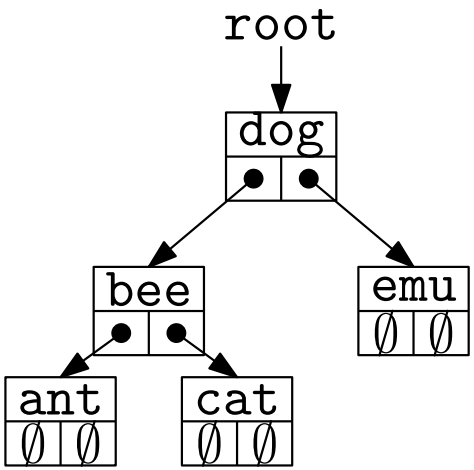
$\forall n \; 2^h \le n < 2^{h+1}, \; H(n) = h$

$h \le \lg n$

$h > \log n - 1$

$h = \lfloor \lg n \rfloor$

Ordered binary tree ADT

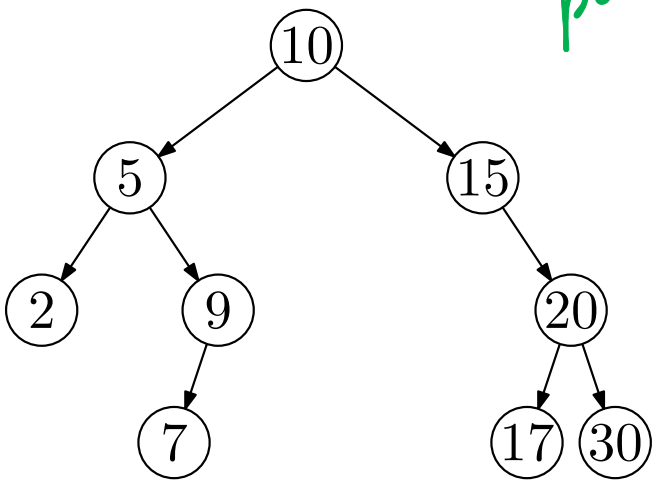


Tree ADT

- insert
- remove
- traverse

```
template<class T>
  class tree {
  public:
  ...
  private:
    struct Node {
      T data;
      Node * left;
      Node * right;
    };
    Node * root;
   ...
};
```

Traversals



postorder

cout << x->data;

← pre order

```
inOrder( Node * x ) {
  If (x != null) {
    inorder(x->left);
    inorder(x->right);
  }}
```

In order: 2, 5, 7, 9, 10, 15, 17, 20, 30

sorted if tree BST

LVR

preorder: 10 5 2 9 7 15 20 17 30

VLR

RVL ni

postorder:

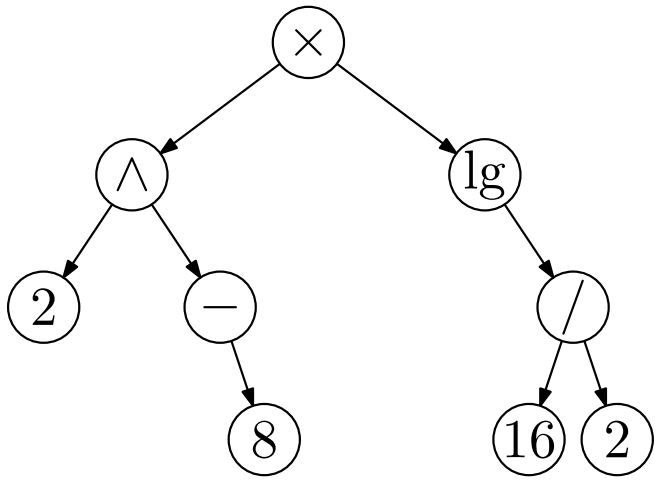
10

RLV erp

LRV

VRL tsop

Expression Evaluation

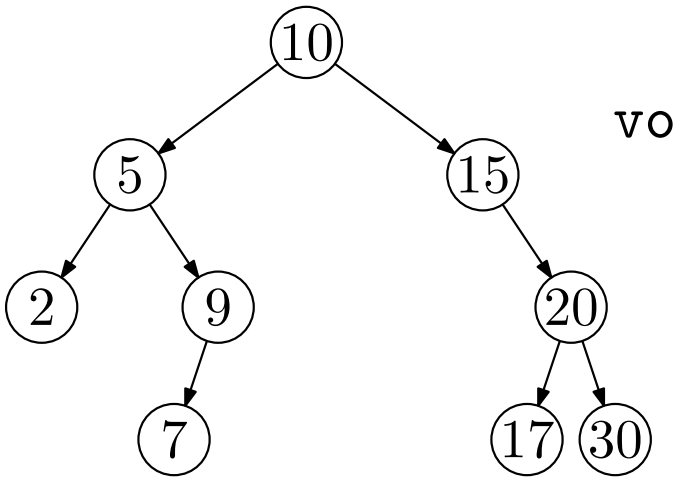



```
double eval( Node * x ) {
  If (x != null) {

    double a = eval(x->left);

    double b = eval(x->right);

  }}
```

Tree Copy

Tree Clear

Level (Depth) order




```
void levelOrder( ) {
  If( root == NULL) return;


}}
```