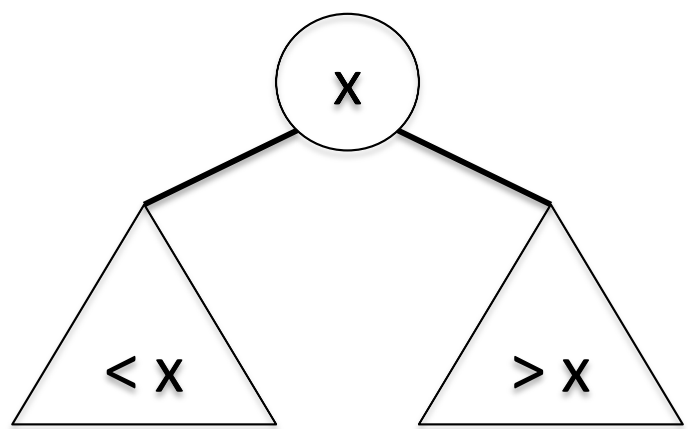
Balanced Binary Search Trees

9/12/24

Recall: Binary Search Trees (BSTs)

- Tree with key stored at each node such that
  – Every node has 2 children (left and right)
    - Children can be null (nodes have 0, 1, 2 actual children)
  – All keys in a node's right subtree are greater than its key and all in the left subtree are less

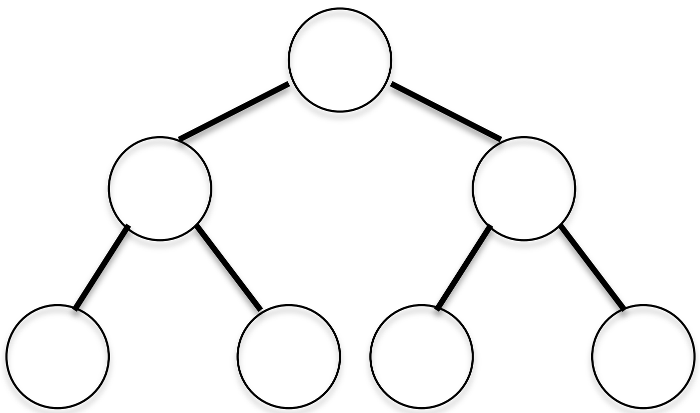


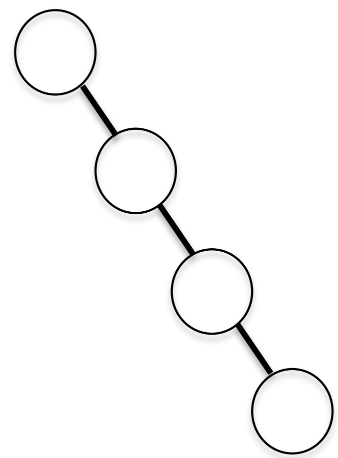
Key property: Height

- BST height: Max # edges on root-to-leaf path
- Insert, delete, contains all run in O(height)

Key property: Height

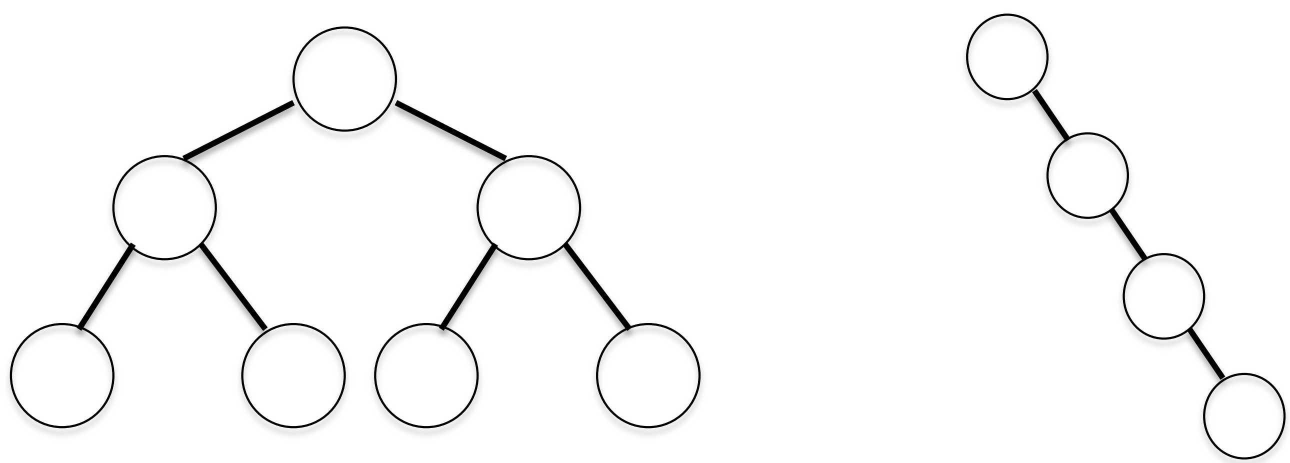
- BST height: Max # edges on root-to-leaf path
- Insert, delete, contains all run in O(height)
- O(log n) ≤ height of n-node BST ≤ n-1:





Key property: Height

- BST height: Max # edges on root-to-leaf path
- Insert, delete, contains all run in O(height)
- O(log n) ≤ height of n-node BST ≤ n-1:



- A tree is *balanced* if its height is O(log n)

Full binary tree

- Every interior node (i.e. non-leaf) has 2 children and all leaves are on the same level

Full binary tree

- Every interior node (i.e. non-leaf) has 2 children and all leaves are on the same level

How many nodes are in a full binary tree of height 3?

A. 7
B. 13
C. 16
D. 27
E. None of the above

Full binary tree

- Every interior node (i.e. non-leaf) has 2 children and all leaves are on the same level

How many nodes are in a full binary tree of height 3?

A. 7
B. 13
C. 16
D. 27
E. None of the above (15)

Claim: A full binary tree of height h has $2^{h+1}-1$ nodes

Claim: A full binary tree of height h has 2^{h+1}-1 nodes

Corollary: A binary tree with n nodes has height $\Omega(\log n)$

Claim: A full binary tree of height h has 2^{h+1}-1 nodes

Corollary: A binary tree with n nodes has

height $\Omega(\log n)$

Pf: $n <= 2^{h+1} - 1$

$n+1 <= 2^{h+1}$

$\log_2 (n+1) <= \log_2 2^{h+1} = h+1$

$h >= \log_2 (n+1) - 1 = \Omega(\log n)$

Induction framework

- Base case(s):
  - claim is true for one or more small values
- Induction hypothesis (IHOP)
  - assume claim is true up to some point
- Induction step
  - show that claim holds for next value (using IHOP)

Prove that $1+2+3+...+n = n(n+1)/2$

- Be sure to:
  - Clearly mark the IHOP and when it’s used
  - Work from one side (typically the left) and get the other side; don’t perform operations on both sides

Prove using induction: A full binary tree of height h has $2^{h+1}-1$ nodes

Use induction to prove that a binary tree with n (≥ 1) nodes has n-1 edges

Use induction to prove that a binary tree with n (≥ 1) nodes has n-1 edges

Need strong induction: Assume claim holds for all smaller values

Fibonacci numbers

- $F_0 = 0$
- $F_1 = 1$
- $F_n = F_{n-1} + F_{n-2}$

Fibonacci numbers

- $F_0 = 0$
- $F_1 = 1$
- $F_n = F_{n-1} + F_{n-2}$

$$F_n = \frac{1}{\sqrt{5}}\left(\left(\underbrace{\frac{1+\sqrt{5}}{2}}_{\phi}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right)$$

AVL trees

Height of an arbitrary node

- Height of a node is the height of the subtree rooted at that node

AVL tree

[Adelson-Velsky and Landis, 1962]

- left height = height if first step must go left
- right height = height if first step must go right
- balance of a node: left height – right height

AVL tree

[Adelson-Velsky and Landis, 1962]

- left height = height if first step must go left
- right height = height if first step must go right
- balance of a node: left height – right height

- AVL tree: Every node has balance in {0,+1,-1}

w_h = # nodes in smallest AVL tree of height h

w_h = # nodes in smallest AVL tree of height h

What is w_4?

A. 9
B. 12
C. 15
D. 18
E. None of the above

w_h = # nodes in smallest AVL tree of height h

What is w_4?

A. 9
B. 12
C. 15
D. 18
E. None of the above

w_h = # nodes in smallest AVL tree of height h

$w_0 = 1$
$w_1 = 2$
$w_h = 1 + w_{h-1} + w_{h-2}$

w_h = # nodes in smallest AVL tree of height h

$w_0 = 1$
$w_1 = 2$
$w_h = 1 + w_{h-1} + w_{h-2}$

Claim: $w_h = F_{h+3} - 1$

w_h = # nodes in smallest AVL tree of height h

$w_0 = 1$

$w_1 = 2$

$w_h = 1 + w_{h-1} + w_{h-2}$

Claim: $w_h = F_{h+3} - 1$

$$> \frac{\phi^{h+3}}{\sqrt{5}} - 2$$

$$F_n = \frac{1}{\sqrt{5}}\left(\left(\underbrace{\frac{1+\sqrt{5}}{2}}_{\phi}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right)$$

w_h = # nodes in smallest AVL tree of height h

$w_0 = 1$
$w_1 = 2$
$w_h = 1 + w_{h-1} + w_{h-2}$

Claim: $w_h = F_{h+3} - 1$

$$> \frac{\phi^{\,h+3}}{\sqrt{5}} - 2$$

$$h = O(\log_{\phi} n)$$

$$\phi$$

$$F_n = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right)$$

w_h = # nodes in smallest AVL tree of height h

$w_0 = 1$

$w_1 = 2$

$w_h = 1 + w_{h-1} + w_{h-2}$

Claim: $w_h = F_{h+3} - 1$

$$> \frac{\phi^{h+3}}{\sqrt{5}} - 2$$

$h = O(\log_\phi n)$

$h = O(1.44\ldots \log_2 n)$

$$F_n = \frac{1}{\sqrt{5}}\left(\left(\underbrace{\frac{1+\sqrt{5}}{2}}_{\phi}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right)$$