**CIS 263 – Week 2: *Trees***

**Trees**

* Special type of graph A graph with no cycles (i.e., acrylic graph)
* Is this a tree?

**A picture containing screenshot, text, circle, exercise

Description automatically generated**

* No, it has a cycle.
* Is this a tree?

A picture containing screenshot, circle, clock

Description automatically generated

* Yes, there are no cycles.

**Binary Search Tree**

* Properties
  + At most 2 children
  + The keys of a node satisfy the following property:
    - Let x be a nod in a binary search tree. If y is a node in the left subtree of x, then y.key <= x.key. If y is a node in the right subtree of x, then y.key >= x.key.
* Is this a binary search tree?

**A screenshot of a computer

Description automatically generated with low confidence**

* Yes, because values to the left of 12 are less than 12, opposite for values right of 12.
* Inorder Tree Walk:

A picture containing text, font, white, design

Description automatically generated

* What order are we printing a tree out?
* Searching a binary search tree?
  + We can find things faster.

A screenshot of a computer

Description automatically generated with low confidence

* This is an iterative tree search.

A picture containing text, font, handwriting, white

Description automatically generated

* Finding the minimum in a BST
  + Just keep going down the left of the tree until you get the min.
* Finding the max of a BST
  + Keep going down the right side to get the max.
* Inserting a node:

**A screenshot of a computer

Description automatically generated with medium confidence**

* Finding a successor in a BST.
  + The successor of node x is the node with the smallest key greater that x.key.

**Red - Black Trees:**

* Introduction
  + Special type of tree with one extra storage bit per node that is red/black.
  + The goal of this is to ensure that no such path is twice as long as any other, so that the tree is approximately balanced.
  + We want to avoid trees like this:

**A screen shot of a graph

Description automatically generated**

* + Each node will contain the following things:
    - Key
    - Left
    - Right
    - Parent
    - NEW: color
  + There is a new null node.
  + The red/black properties include:
    - Every node is red/black.
    - The root is black.
    - Every lead (NIL) is black.
    - If a node is read, then both its children are black.
    - For each node, all simple paths from the node to descendant leaves contain the same number of black nodes.
  + Examples:

**A screenshot of a diagram

Description automatically generated**

**A diagram of a number of black and red circles and black text

Description automatically generated**

* + **Black-height:** the number of black nodes on any simple path from, but not including a node x down to a leaf.

**A diagram of a black and red triangle

Description automatically generated**

* + For each node, all simple paths from the node to descendant leaves contain the same number of black nodes.
  + A red/black tree with n internal nodes has height at most

**Red/Black Tree insertion:**

* We want to insert 12 into the tree shown above.

A screenshot of a computer

Description automatically generated

* If it is red, it breaks a rule, if it is black, it also breaks a rule.

**Rotations:**

**A screenshot of a computer

Description automatically generated**

* This will be helpful for fixing the color of the red/black tree.