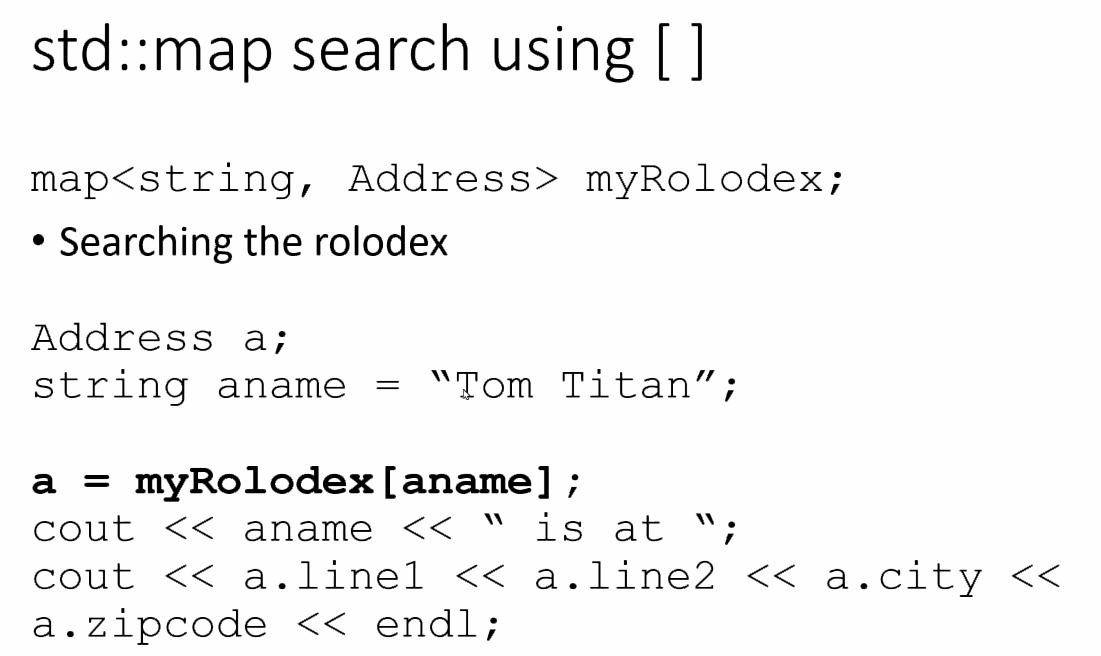
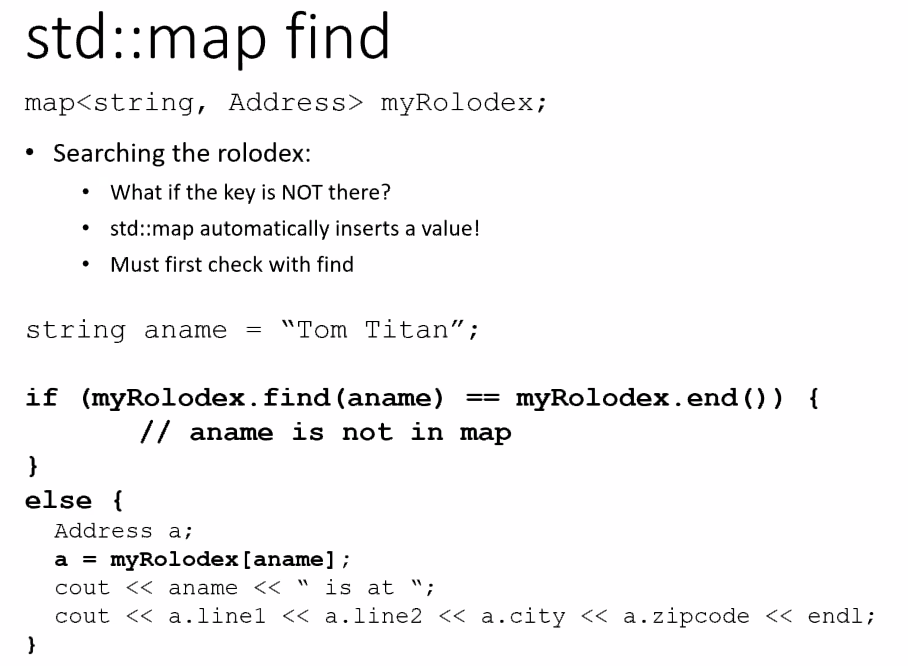
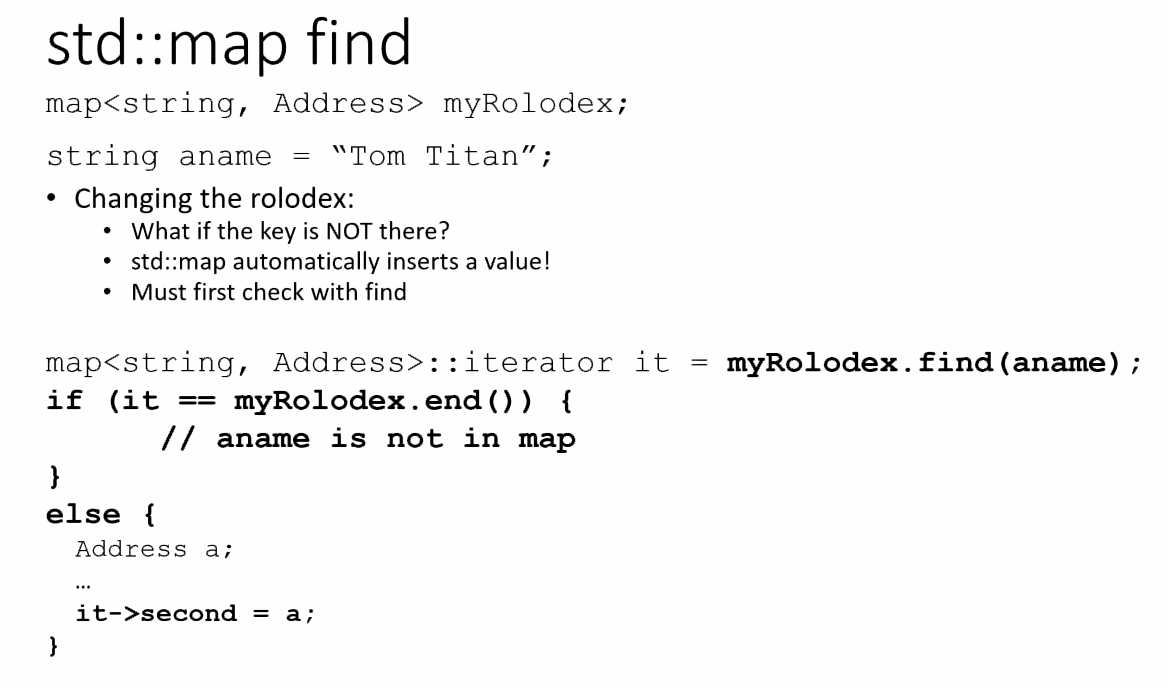
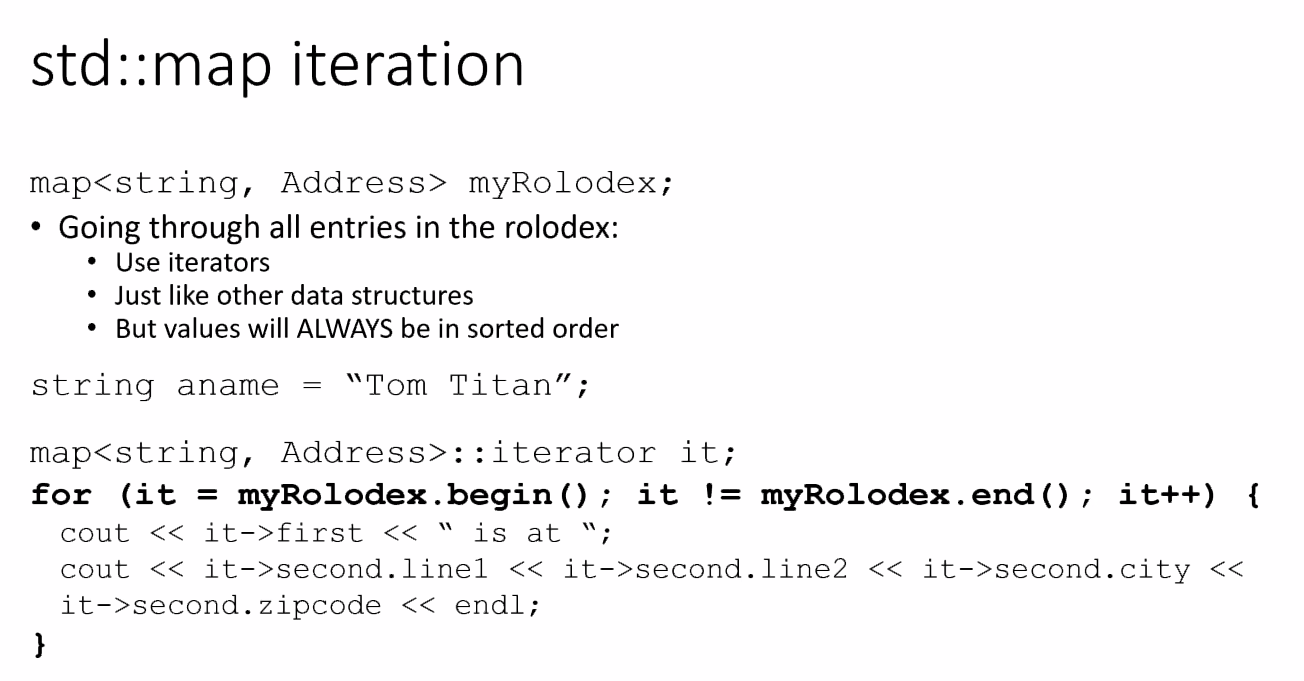
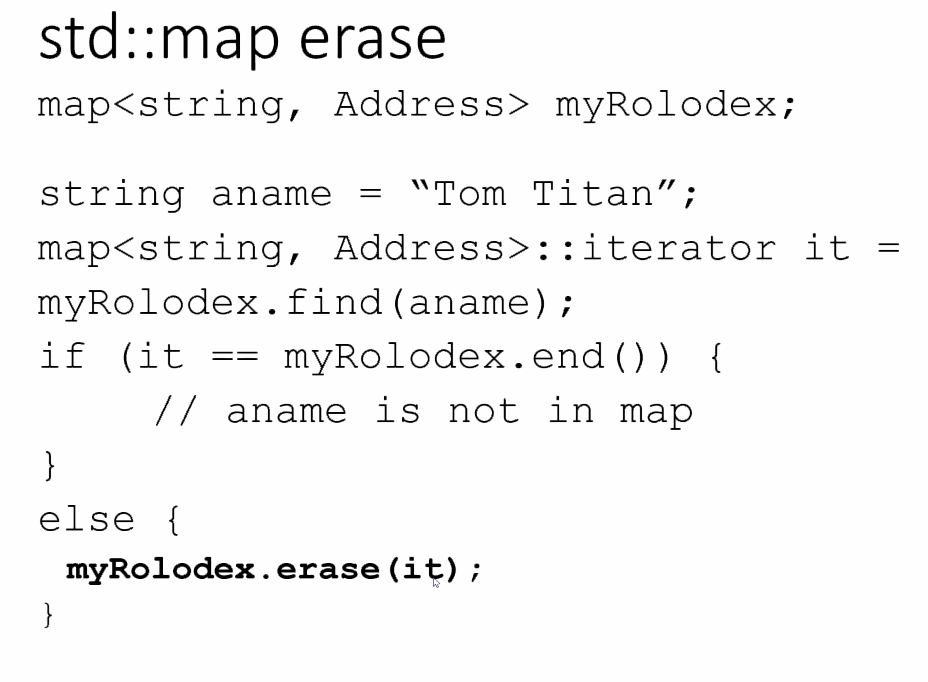
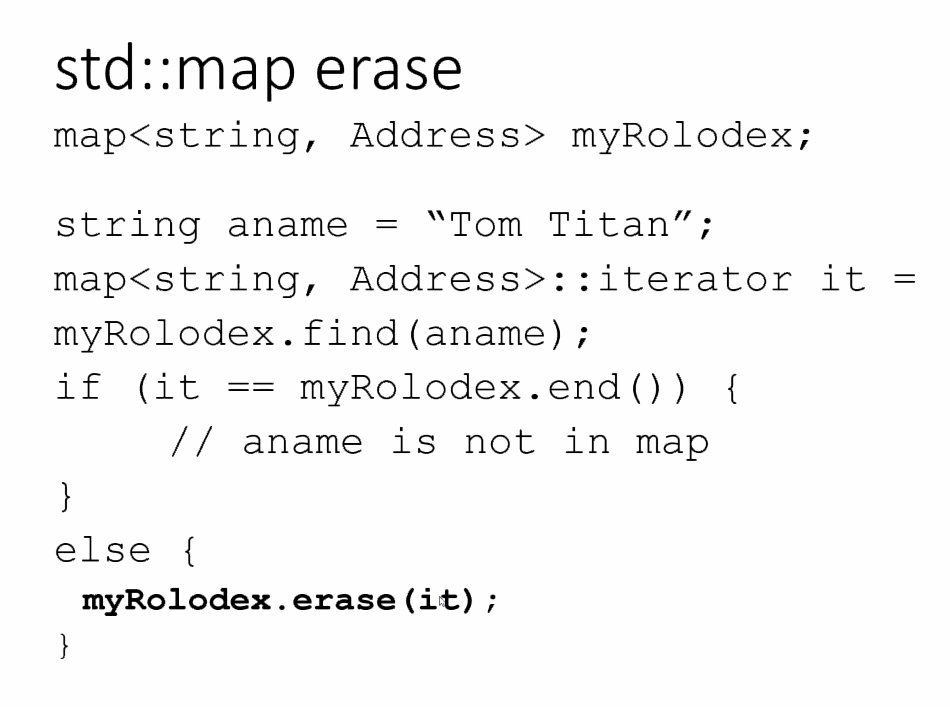
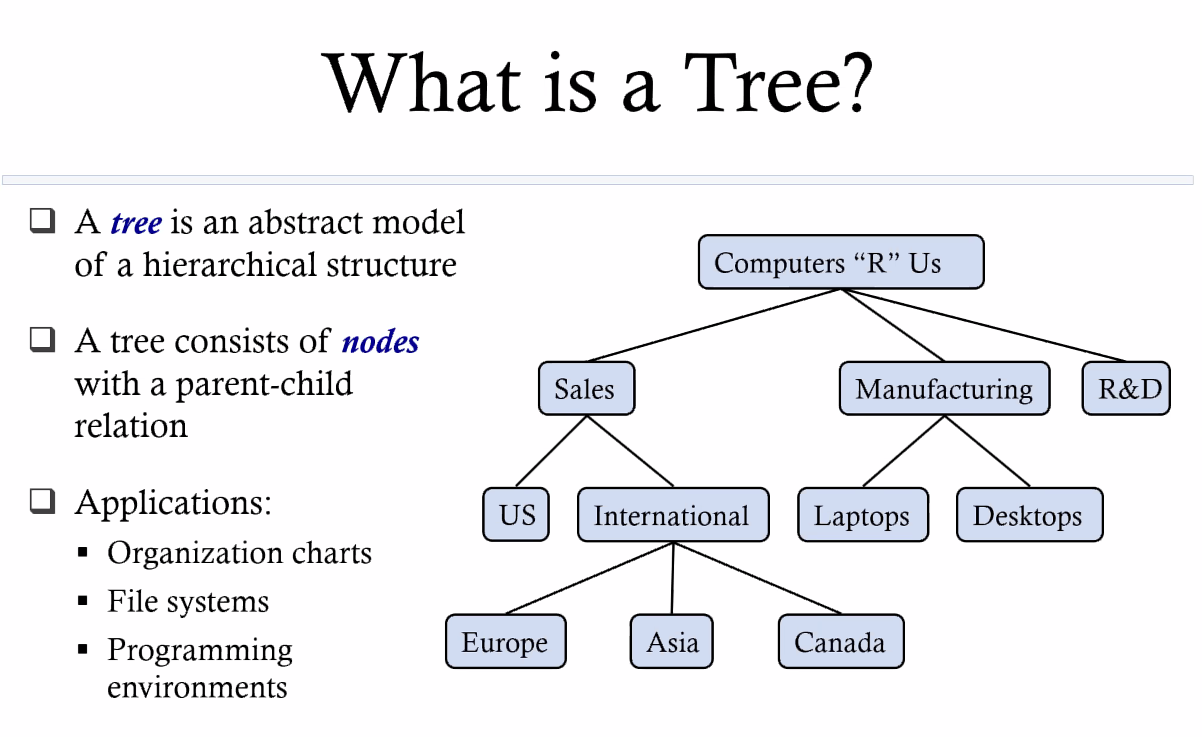
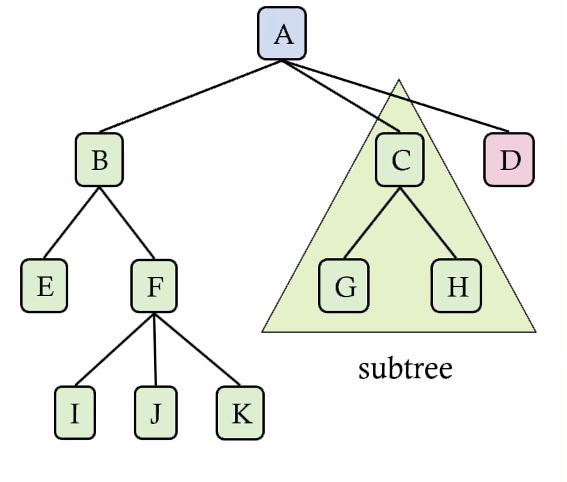
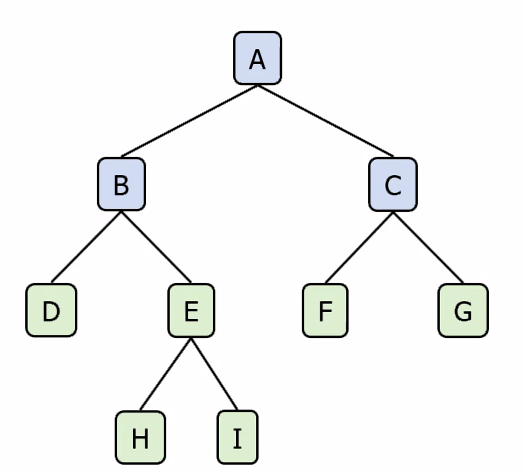
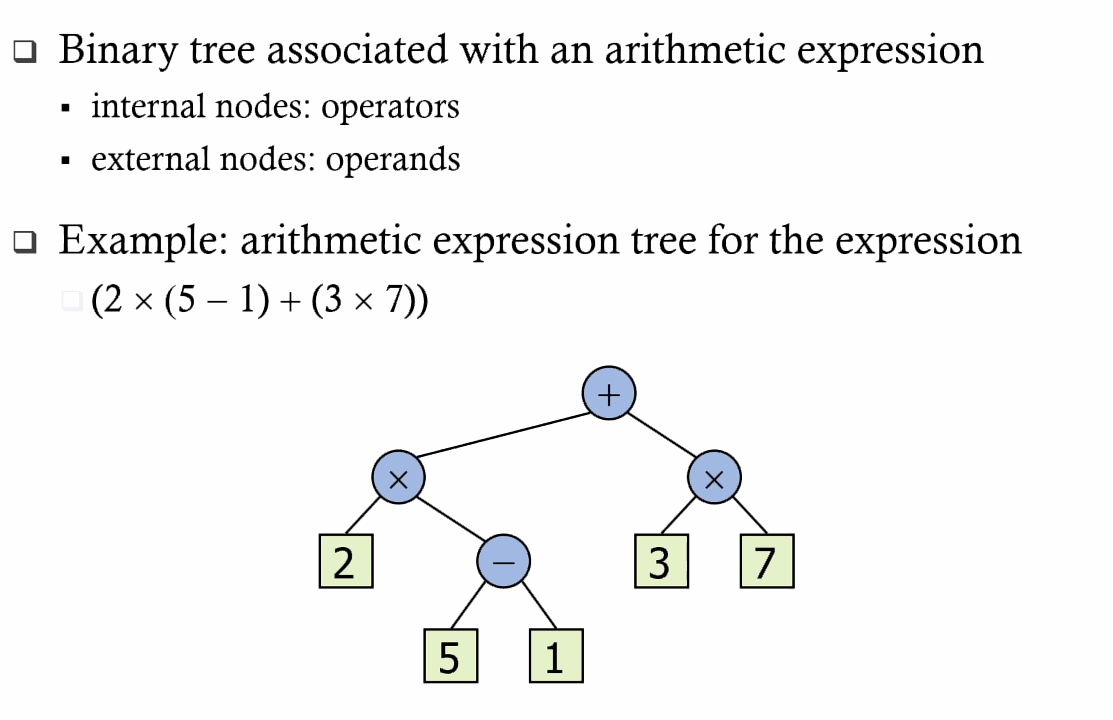
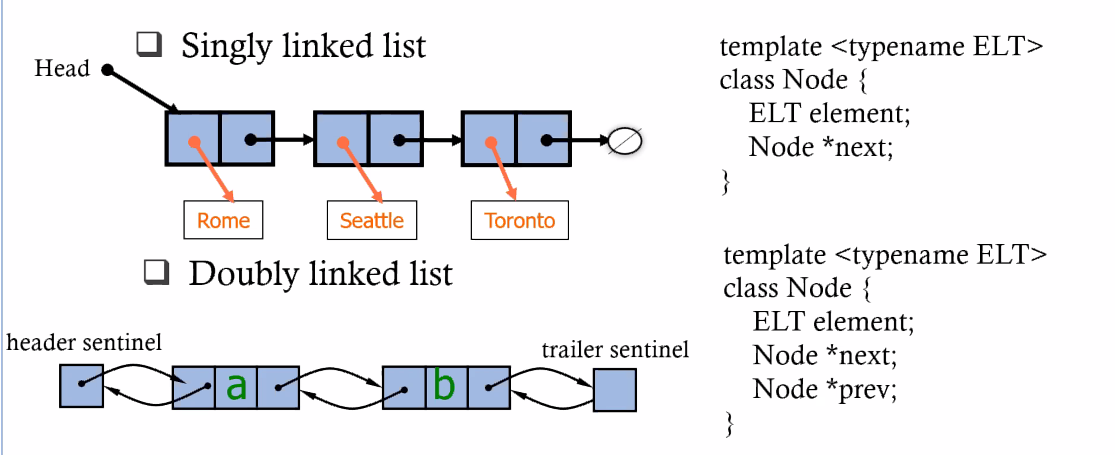
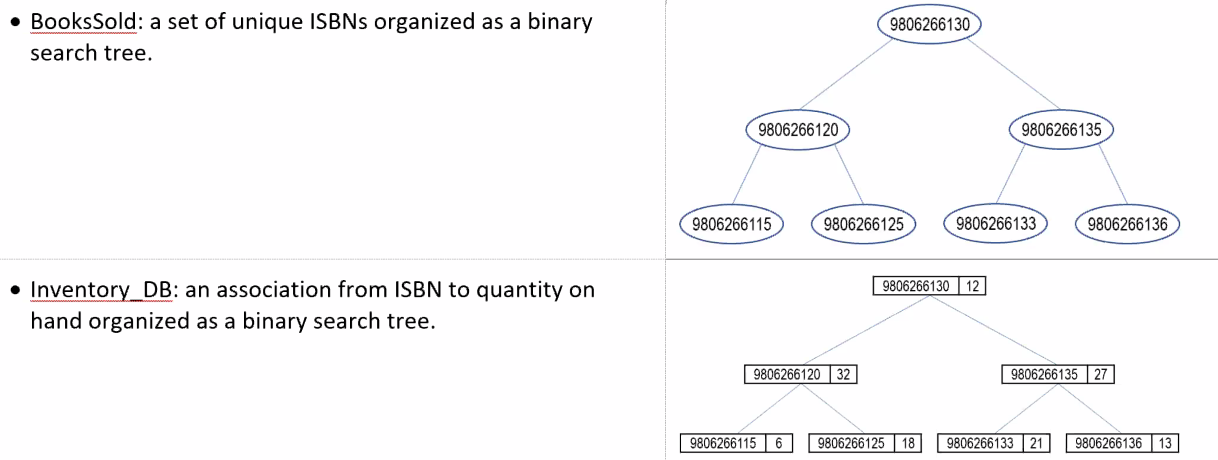
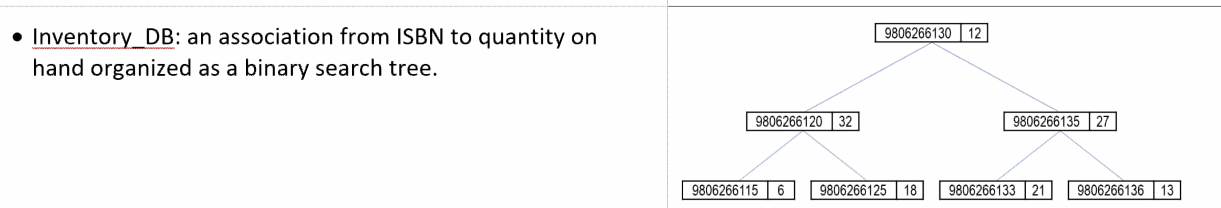
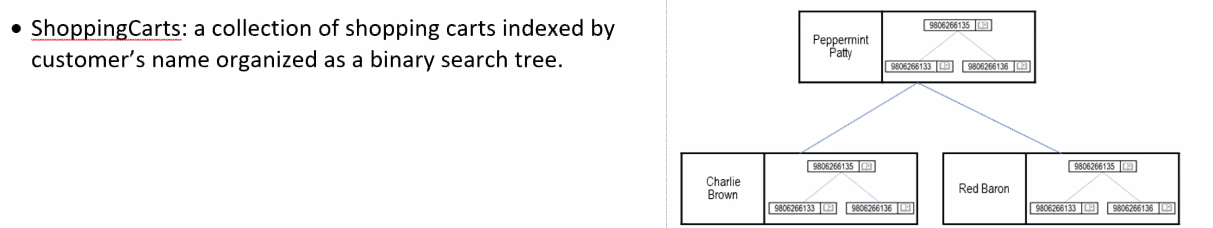
Lecture 16

CPSC 131  
10/28/2020

1. Std::map search  
   
2. Std::map find  
   
   1. This helps you find a thing without inserting it into the tree
   2. Helps you find it in log2n time
3. A better code  
   
   1. It points to a value not the node and it introduces you to the second value.
4. Std::map iteration  
   
5. Erase  
   
   1. This is a O(1)
   2. As long as you know hwere, you can do it
6. Erase (with a key)  
   
   1. Erase will have to serach and define the node
7. Trees
   1. What is a tree  
      
   2. Tree Terminology  
      
      1. Root: the head of the tree
      2. Internal node: node with at least one child (A,B,C,F)
      3. External node (aka leaf): node without children (E,I,J,K,G,H,D)
      4. Depth of a node: number of ancestors (between the node and the root. Root has depth 0)
      5. Height of a tree: maximum depth
      6. Descendant of a node: child, grandchild, grand-grandchild, etc.
      7. Subtree: tree consisting of a node and its descendants
      8. Sibling: Nodes that share a parent; nodes at the same level
   3. Binary Trees  
      
      1. Binary tree is a tree with the following properties:
         1. Each internal node has at most two children
         2. Children of a node are an ordered pair
      2. We all children of an internal node: left and right
      3. Types of binary trees
         1. Full: if every node has 0 or 2
         2. Complete: All levels are full except possibly the last level
            1. Example: look at Row 2 (DEFG)
         3. Perfect: All internal nodes have 2 children and leaf nodes are at the same level
      4. Applications:
         1. Arithmetic expressions
         2. Decision processes
   4. Arithmetic Expression Tree
      1. 
   5. Decision Tree
      1. IF (want coffee) -> go to Starbucks
   6. Defining a Node in C++  
      
      1. Discuss later
8. Project4
   1. Last time, it was a BookDatabase with which we used to check out and purchase books
   2. Now, we’re building the store.
   3. Each customer has a cart with their own books. Only we aren’t focused on one customer but multiple carts.
   4. Data Types of BookStore.cpp  
      1. BooksSold  
         
         1. This needs to be a Binary Tree. We know this to be a *set*
      2. Inventory  
         
         1. Isbn() and quantity of books
         2. Binary search tree
      3. ShoppingCart  
         
         1. Binary search tree
         2. Indexed by isbn
      4. ShoppingCarts (plural)  
         
         1. Binary search tree
         2. Indexed by customer name
   5. Main.cpp and its Data Types
      1. Create store, make shoppers, make books, re-order inventory
      2. If number of books dip below a threshould, reorder more
   6. BookDatabase
      1. We’re going to reuse it from the last time except we have to build a tree in it