

CSE12 DISCUSSION 7



Pooja Bhat

Binary Search Trees

- Hierarchical data structure with a single reference to root node
- BST Property: The key of any node is greater than all keys occurring in its left subtree and less than all keys occurring in its right subtree.
- The depth of a node is the number of edges from the root to the node.
- The height of a node is the number of edges from the node to the deepest leaf.
- The height of a tree is a height of the root.
- A full binary tree is a binary tree in which each node has exactly zero or two children.
- A complete binary tree is a binary tree, which is completely filled, with the possible exception of the bottom level, which is filled from left to right.

Insertion

Insert 25, 15, 10, 50, 35, 22, 24, 4, 18, 70, 12, 31, 90, 66

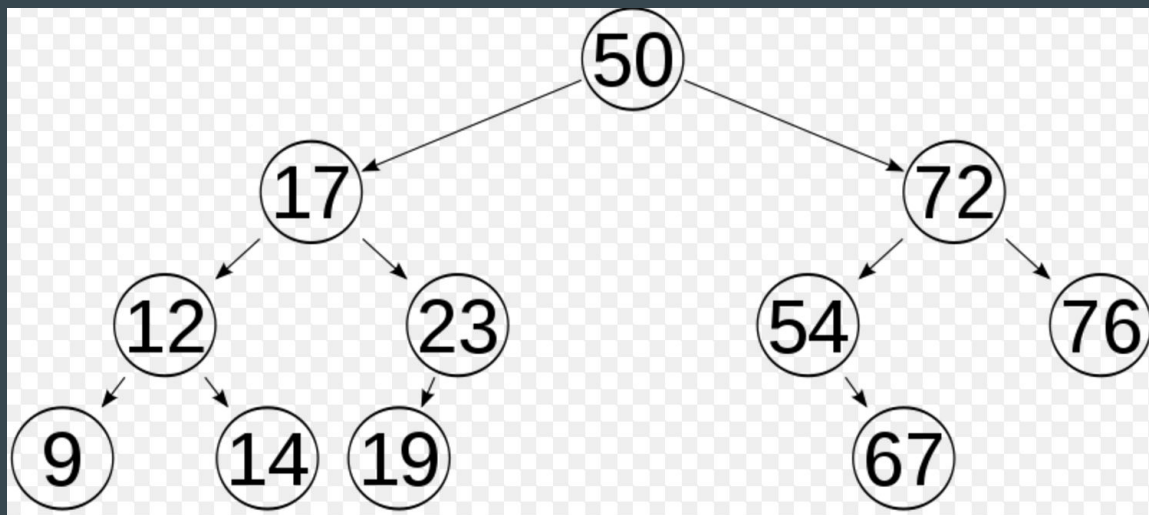
- What is the height of the tree?
 - What is the depth of the root node?
 - What is the maximum depth in the tree?
 - Is it a complete tree?
 - Is it a full binary tree?
-
- Notice that a new item is always inserted at the leaf node, unlike a binary heap
 - Insert, Search and remove is all $O(\log n)$

Tree Traversals

- Inorder Traversal - Left, Root, Right
- Preorder Traversal - Root, Left, Right
- Postorder Traversal - Left, Right, Root

BST Iterator

- How to iterate over a tree?
- We basically implement the inorder traversal algorithm iteratively using a stack
- Consider the BST below: What is the inorder traversal of the tree?

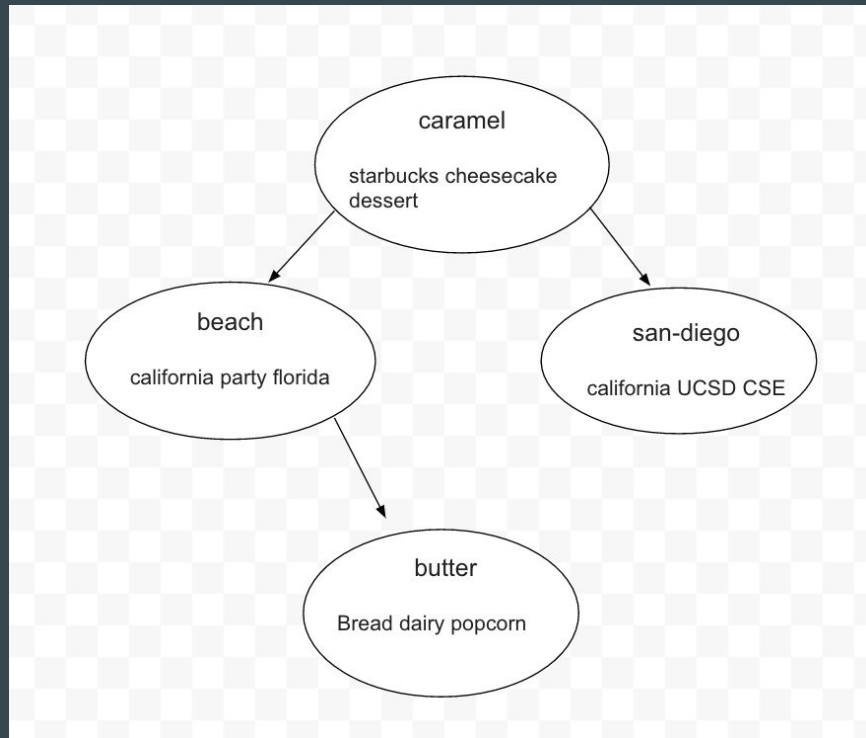


BST Iterator

Content of Stack in
the constructor: (top
to bottom) : 9, 12, 17,
50

Call to next()	Value Returned	Content of Stack after the call
1	9	12, 17, 50
2	12	14, 17, 50
3	14	17, 50
4	17	19, 23, 50
5	19	23, 50
6	23	50
7	50	54, 72
8	54	67, 72
9	67	72
10	72	76
11	76	-

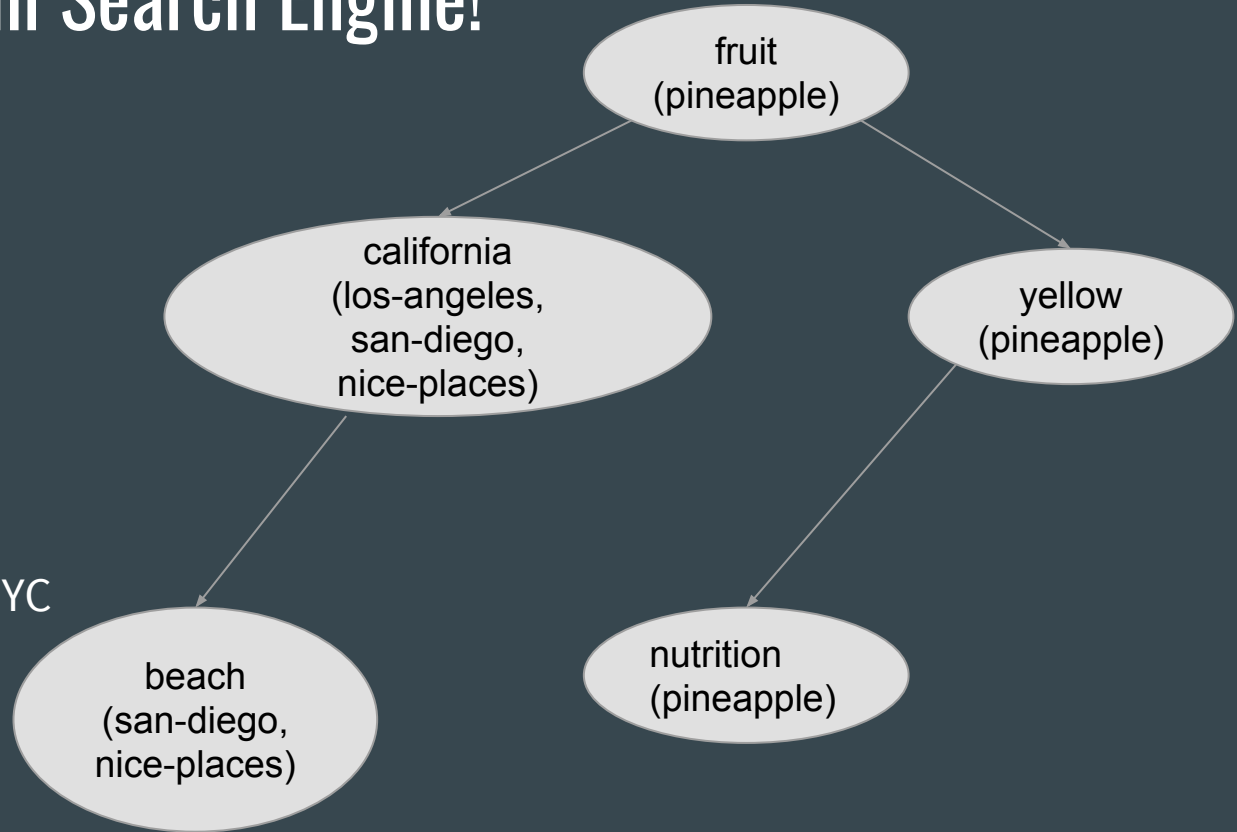
HW7 Part 1



- Each node in a BST has a key<T> and a LinkedList<T>
- For the purpose of Part 1, it does not matter what information you store in the LinkedList
- Implement the methods insert, search, etc.
- At least 3 methods must be done recursively
- Use private helper methods to do it recursively.
- Use the iterator to test you tree. Remember that inorder traversal returns the items in sorted order.

Part 2 - Your mini Search Engine!

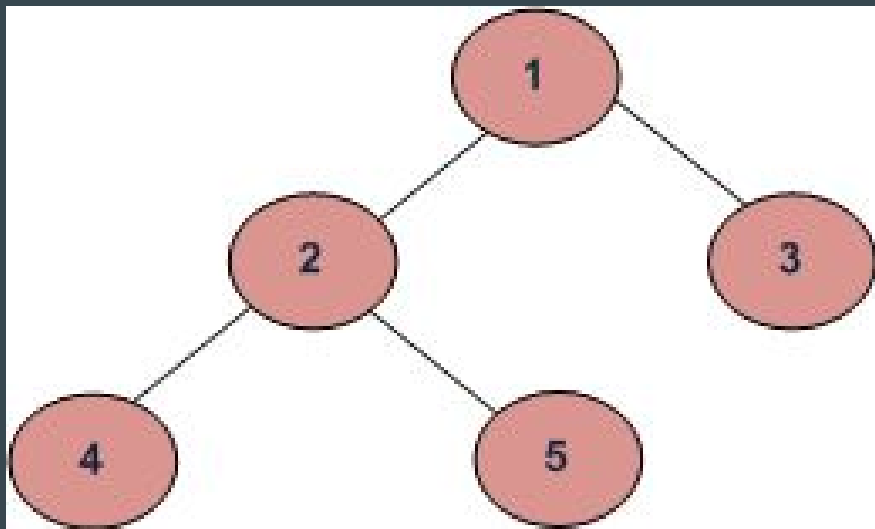
Pineapple
fruit yellow nutrition
Los-Angeles
California sunny warner
San-Diego
California beach UCSD
Nice-Places
California sunny beach NYC



A part of your BST for the above file would look like the tree above.

Interview Questions

Given a BST, write a method that prints the level order traversal of the tree
Level order traversal of the tree below: 1, 2, 3, 4, 5



Interview Questions

Given a BSTree tree, validate that it satisfies the BST property.

I.e. The key of any node is greater than all keys occurring in its left subtree and less than all keys occurring in its right subtree.

