

# Implementing a Binary Search Tree and a Binary Tree Search Algorithm

## Homework #4

By Logan Miles

### 1. Objectives

The goal of this assignment was to write a program that would read a CSV file and implement an algorithm that would add the items from the CSV into a binary search tree with respect to each item's unique UPC key and the binary search tree properties. Another algorithm was required to traverse the binary search tree using in-order traversal, printing the attributes of the item at each node including its UPC key, amount, and name. The last algorithm that was to be implemented was supposed to use the binary search tree properties to locate an item in the tree with a specific UPC key. Each item's UPC key, along with the item's amount and name, could be found in another file titled "input.dat" which also needed to be read in order to compare the key of the target item to keys of those in the binary search tree.

### 2. Program Design

Because this assignment required the implementation of a binary search tree and Java is an object oriented programming language, the first thing implemented was a `TreeNode` class that represented the attributes of the items in the CSV file. A `BinaryTree` class was then created that contains the necessary functions to create the binary search tree, print each node using in-order traversal, and search the tree for a particular key. Lastly, the `HW4` class

contains the main() function that is used to call the functions from the other classes and test the algorithms. The following functions are contained within these classes:

### **TreeNode()**

TreeNode() is a basic constructor for the TreeNode class that establishes each object's key, amount, name, and parent. The amount and name are of the String type, while the key is of the type Long, and the parent (or p) of the node is of the type TreeNode as well. The amount is a String instead of an integer because the amounts of each item in the csv file are of different types and most of them contain alphabetical characters. These amounts also do not need to be operated on, only printed. Lastly, all keys within these classes are of the type Long because there are items in the csv file with a UPC key larger than the maximum value possible for a signed 32-bit integer. The type of unsigned integer also would have worked as a solution, as it also can represent values above 2,147,483,647.

### **insert()**

Constructs the binary search tree by inserting each TreeNode object created based on the attributes read from the UPC.csv file. Each node is inserted with respect to its unique UPC key and the binary search tree properties. An iterative solution had to be implemented using while loop's because a recursive solution would result in a stack overflow at runtime due to the function hitting recursion depth. The function determines the correct place for the node to be placed in the tree by iterating through the tree with a while loop, testing an if condition with each loop. The loop continues if the current node is not null, meaning the loop will stop once it has reached a leaf. The if condition tests if the key of the node to be added is less than that of the current node. If so, it sets the current node to the left node,

and if not, it sets it to the right node. A second set of if statements outside the while loop tests determine if the node to be added is the left or right child of this node. The first if statement tests if the parent of the node to be added is null, making it the root if so. The second tests if the parent's key is less node's key, making it the left child if so and the right child if not. This ensures that less than the node to be added is to the left and any node greater than the node is on the right, following the binary search tree property.

### **printInOrderTraversal()**

Uses in-order traversal to visit every node contained in the binary search tree. This code uses a stack to simulate the recursive in-order traversal of a binary search tree. The reason that a recursive solution could not be implemented is because it would result in a stack overflow at runtime due to the function hitting recursion depth. The function starts from the leftmost node and works its way up to the root while printing the nodes in ascending order of their keys. Before the in-order traversal, an if statement tests if the tree is empty and returns null if so, preventing the code from crashing. The function first initializes this stack then continues a while loop if the current node is not null and the stack is greater than zero. A nested while loop traverses the leftmost subtree and pushes it onto the stack. After the nested while loop exits the current node is popped from the stack, and a series of if statements print the node's key, amount if applicable, and name. Finally, the current node is set to the right child to ensure that the right children nodes are printed as well.

### **searchTree()**

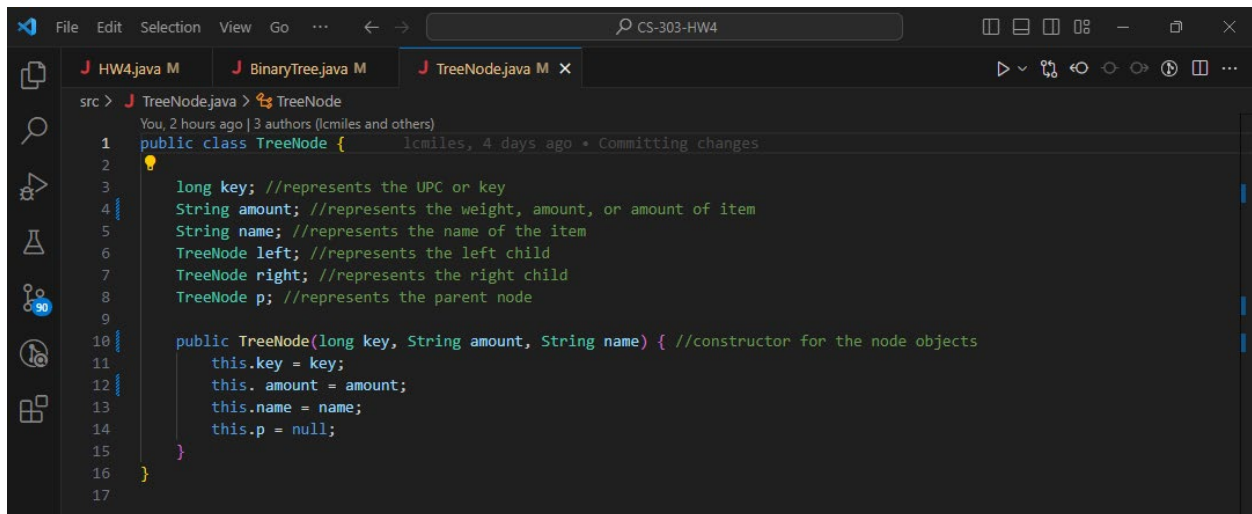
Takes advantage of the already established binary search tree properties to locate a specific node using its unique UPC key. The function iterates through the tree using a while loop, continuing if the current node is not null and the target key does not match that of the current node's key. An if statement tests if the target key is less than the node's key, assigning the current node to the left node for the next iteration if so, and assigning it to the right node if not. Due to the binary search tree properties, the key must be to the left if it is less and to the right if it is greater, as each node's left child is less and right child is greater. Once the while loop exits, the target key must match the key of the current node, and the function returns this node.

### **main()**

This function reads the UPC.csv and input.dat files and calls the functions necessary to build the binary search tree, print each node using in-order traversal, and search for a particular node's key and print the results. First, the function creates a BinaryTree object so that the other functions can be called on it later. Then a scanner object is initialized and called on the CSV file. A while loop iterates through each line of the CSV file. The resulting string is split at each comma using a regex, and each element of the string array is assigned to an attribute of the TreeNode object and constructed. The key attribute is first converted to a long using the parseLong() function. The TreeNode object is then added to items array list. While the insert() function could have been called on each object directly, it was found to be much slower in testing. The arraylist is then iterated over using a for loop, and the insert() function is called on each object. The printInOrderTraversal() is then called on the

tree to print every node's attributes. Next, a new scanner is instantiated and called on the input.dat file. Using a similar method to before, the resulting string is split at each comma using a regex, the first element of each array is converted to a long, and then added to an array list. A for loop then iterates through this array list, calling the searchTree() function on each key and the root of the tree. A series of print statements then print out the result's key, amount if applicable, and name. An if statement was also added to test if the result is null, ensuring that the code does not crash if the key was not found. The function also records the search and insert functions' execution time and prints it in nanoseconds, milliseconds, and seconds, then prints the results.

### Code Screenshots:

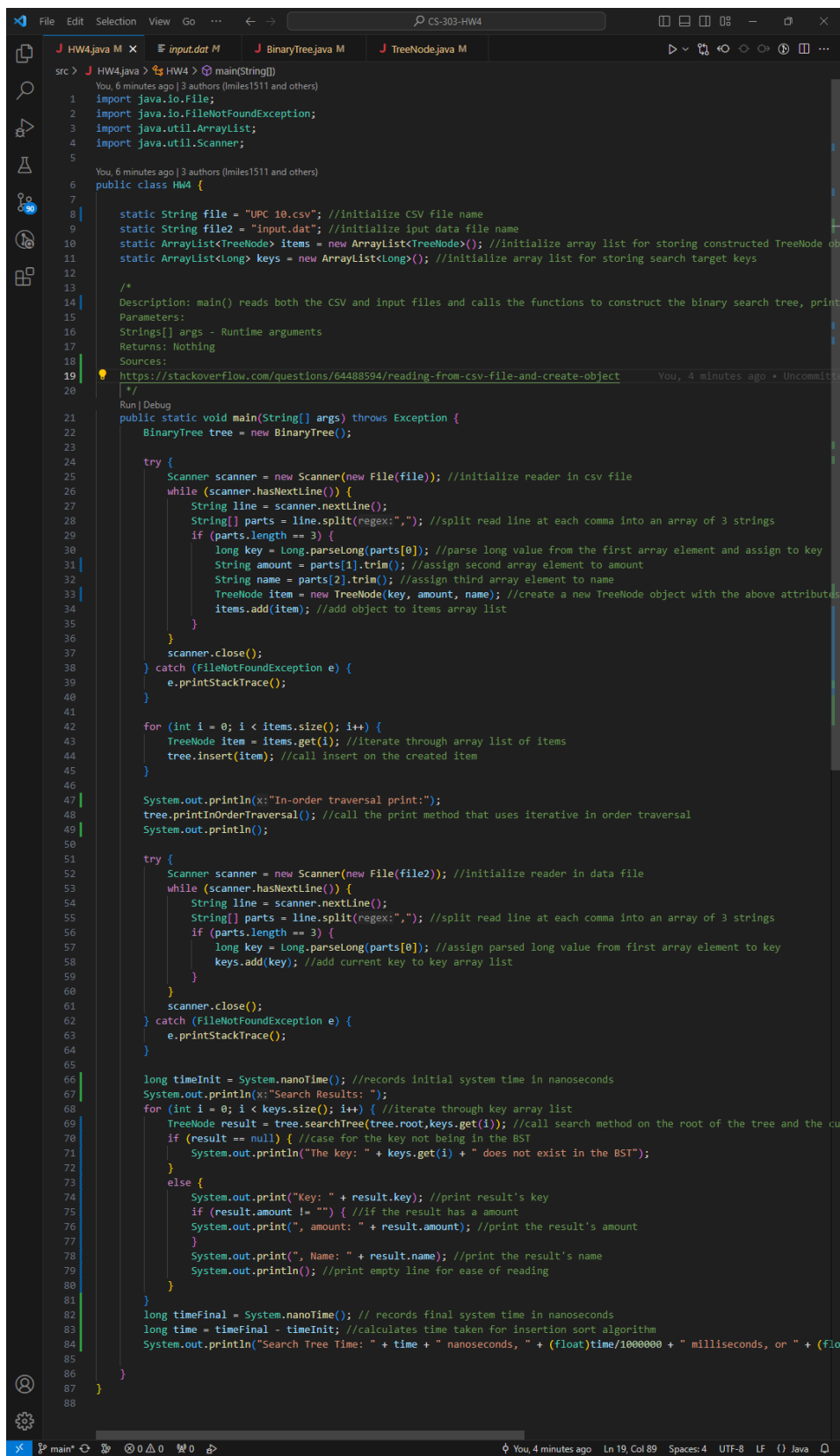


```
src > J TreeNode.java > TreeNode
You, 2 hours ago | 3 authors (icmiles and others)
1 public class TreeNode { icmiles, 4 days ago • Committing changes
2
3     long key; //represents the UPC or key
4     String amount; //represents the weight, amount, or amount of item
5     String name; //represents the name of the item
6     TreeNode left; //represents the left child
7     TreeNode right; //represents the right child
8     TreeNode p; //represents the parent node
9
10    public TreeNode(long key, String amount, String name) { //constructor for the node objects
11        this.key = key;
12        this.amount = amount;
13        this.name = name;
14        this.p = null;
15    }
16 }
17
```

Figure 1: TreeNode.java

```
File Edit Selection View Go ... CS-303-HW4
J HW4.java M J BinaryTree.java M X J TreeNode.java M
src > J BinaryTree.java > BinaryTree > printInOrderTraversal()
3 public class BinaryTree {
4     TreeNode root; //initialize binary search tree root object
5
6     /*
7     Description: This function adds a node to the tree while maintaining the binary search tree properties
8     Parameters:
9     TreeNode z - the TreeNode object to be added
10    Returns: Nothing
11    Sources:
12    https://www.youtube.com/watch?v=2gZYhlhYdYA
13    https://chat.openai.com/c/78371716-dd16-491e-a6b8-e9bc5faa34d9
14    */
15    public void insert(TreeNode z) {
16        TreeNode y = null; //initialize temp TreeNode variable
17        TreeNode x = root; //initialize the BST root
18
19        while (x != null) {
20            y = x; //set the value of y to current node
21            if (z.key < x.key) { //if the key to be added is less than the current key
22                x = x.left; //set the current node to the left child
23            }
24            else {
25                x = x.right; //else set the current node to the right child
26            }
27        }
28
29        z.p = y; //set the value of the parent of the node to be added to y, the appropriate position for the node to be added
30        if (y == null) { //if the parent does not exist
31            root = z; //the tree is empty and the node becomes the root
32        }
33        else if (z.key < y.key) { //if the node's key is less than the parent's key
34            y.left = z; //set the node to the left child of the parent
35        }
36        else {
37            y.right = z; //set the node to the right child of the parent
38        }
39    }
40
41    /*
42    Description: This function traverses the tree using in-order traversal and prints each TreeNode object's key, amount (if applicable)
43    Parameters: None
44    Returns: Nothing
45    Sources:
46    https://www.youtube.com/watch?v=2gZYhlhYdYA
47    https://chat.openai.com/c/78371716-dd16-491e-a6b8-e9bc5faa34d9
48    https://www.geeksforgeeks.org/binary-search-tree-traversal-inorder-preorder-post-order/
49    https://www.geeksforgeeks.org/inorder-tree-traversal-without-recursion/ You, 3 minutes ago • Uncommitted changes
50    */
51    public void printInOrderTraversal() {
52        if (root == null) { //if the root is null
53            return; //then the tree is empty and return nothing
54        }
55        Stack s = new Stack(); //initialize a stack
56        TreeNode current = root; //set the current node to root
57        while (current != null || s.size() > 0) { //loop until all nodes have been visited and the stack is empty
58            while (current != null) { //traverse to the leftmost subtree and push the nodes onto the stack
59                s.push(current); //push the current node
60                current = current.left; //set the current node to the left child
61            }
62            current = s.pop(); //pop the current node from the stack
63            System.out.print("Key: " + current.key); //print the current node's key
64            if (current.amount != "") { //if the current node has a amount
65                System.out.print(", amount: " + current.amount); //print the current node's amount
66            }
67            System.out.print(", Name: " + current.name); //print the current node's name
68            System.out.println(); //print empty line for ease of reading
69            current = current.right; //set the current node to the right child
70        }
71    }
72
73    /*
74    Description: search() searches the tree by comparing the target key to the current node's key, moving to the left child if the key is less and the right child if the key is greater
75    Parameters:
76    TreeNode node - The starting node that the search function is called on
77    Long key - The target key that the search function compares each TreeNode object's key to
78    Returns:
79    TreeNode node - The node with the key matching the target key
80    Sources:
81    https://www.youtube.com/watch?v=2gZYhlhYdYA
82    https://chat.openai.com/c/78371716-dd16-491e-a6b8-e9bc5faa34d9
83    https://www.geeksforgeeks.org/properties-of-binary-tree/
84    */
85    public TreeNode searchTree(TreeNode node, Long key) {
86        while (node != null && key != node.key) { //while the current node is not null and the key is not the key of the current node
87            if (key < node.key) { //if the target key is less than key of the current node
88                node = node.left; //set the current node to the left child
89            }
90            else {
91                node = node.right; //set the current node to the right child
92            }
93        }
94        return node; //return the current node after exiting the while loop, meaning it has a matching key
95    }
96}
```

Figure 2: BinaryTree.java



```
src > J HW4.java > HW4 > main(String[])
You, 6 minutes ago | 3 authors (lmiles1511 and others)
1 import java.io.File;
2 import java.io.FileNotFoundException;
3 import java.util.ArrayList;
4 import java.util.Scanner;
5
6 public class HW4 {
7
8     static String file = "UPC 10.csv"; //initialize CSV file name
9     static String file2 = "input.dat"; //initialize input data file name
10    static ArrayList<TreeNode> items = new ArrayList<TreeNode>(); //initialize array list for storing constructed TreeNode objects
11    static ArrayList<Long> keys = new ArrayList<Long>(); //initialize array list for storing search target keys
12
13    /*
14     Description: main() reads both the CSV and input files and calls the functions to construct the binary search tree, print
15     Parameters:
16     Strings[] args - Runtime arguments
17     Returns: Nothing
18     Sources:
19     https://stackoverflow.com/questions/64488594/reading-from-csv-file-and-create-object You, 4 minutes ago + Uncommite
20     */
21    Run | Debug
22    public static void main(String[] args) throws Exception {
23        Binarytree tree = new Binarytree();
24
25        try {
26            Scanner scanner = new Scanner(new File(file)); //initialize reader in csv file
27            while (scanner.hasNextLine()) {
28                String line = scanner.nextLine();
29                String[] parts = line.split(regex:","); //split read line at each comma into an array of 3 strings
30                if (parts.length == 3) {
31                    long key = Long.parseLong(parts[0]); //parse long value from the first array element and assign to key
32                    String amount = parts[1].trim(); //assign second array element to amount
33                    String name = parts[2].trim(); //assign third array element to name
34                    TreeNode item = new TreeNode(key, amount, name); //create a new TreeNode object with the above attributes
35                    items.add(item); //add object to items array list
36                }
37            }
38            scanner.close();
39        } catch (FileNotFoundException e) {
40            e.printStackTrace();
41        }
42
43        for (int i = 0; i < items.size(); i++) {
44            TreeNode item = items.get(i); //iterate through array list of items
45            tree.insert(item); //call insert on the created item
46        }
47
48        System.out.println("In-order traversal print:");
49        tree.printInOrderTraversal(); //call the print method that uses iterative in order traversal
50        System.out.println();
51
52        try {
53            Scanner scanner = new Scanner(new File(file2)); //initialize reader in data file
54            while (scanner.hasNextLine()) {
55                String line = scanner.nextLine();
56                String[] parts = line.split(regex:","); //split read line at each comma into an array of 3 strings
57                if (parts.length == 3) {
58                    long key = Long.parseLong(parts[0]); //assign parsed long value from first array element to key
59                    keys.add(key); //add current key to key array list
60                }
61            }
62            scanner.close();
63        } catch (FileNotFoundException e) {
64            e.printStackTrace();
65        }
66
67        long timeInit = System.nanoTime(); //records initial system time in nanoseconds
68        System.out.println("Search Results: ");
69        for (int i = 0; i < keys.size(); i++) { //iterate through key array list
70            TreeNode result = tree.searchTree(tree.root, keys.get(i)); //call search method on the root of the tree and the cu
71            if (result == null) { //case for the key not being in the BST
72                System.out.println("The key: " + keys.get(i) + " does not exist in the BST");
73            }
74            else {
75                System.out.print("Key: " + result.key); //print result's key
76                if (result.amount != "") { //if the result has a amount
77                    System.out.print(", amount: " + result.amount); //print the result's amount
78                }
79                System.out.print(", Name: " + result.name); //print the result's name
80                System.out.println(); //print empty line for ease of reading
81            }
82        }
83        long timeFinal = System.nanoTime(); // records final system time in nanoseconds
84        long time = timeFinal - timeInit; //calculates time taken for insertion sort algorithm
85        System.out.println("Search Tree Time: " + time + " nanoseconds, " + (float)time/1000000 + " milliseconds, or " + (flo
86    }
87 }
88
```

Figure 3: HW4.java

### 3. Testing

The algorithms in this program that were tested include the insert() function, which is the algorithm that builds the binary search tree, and the searching algorithm contained in searchTree(). Both functions were tested using several different versions of the same CSV file, each containing a different number of elements:

- 1) UPC 10 - 10 elements
- 2) UPC 100 - 100 elements
- 3) UPC 1K - 1000 elements
- 4) UPC 10K - 10000 elements
- 5) UPC 100K - 100000 elements
- 6) UPC - The Original CSV file containing 177650 elements

#### Screenshots of testing outputs:

```
PS C:\Users\logan\OneDrive - UAB - The University of Alabama at Birmingham\UAB Files\FA 2023\CS-303\HW\CS-303-HW4> c:; cd 'c:\Users\logan\OneDrive - UAB - The University of Alabama at Birmingham\UAB Files\FA 2023\CS-303\HW\CS-303-HW4'; & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\logan\OneDrive - UAB - The University of Alabama at Birmingham\UAB Files\FA 2023\CS-303\HW\CS-303-HW4\bin' 'HW4'
In-order traversal print:
Key: 79, Name: INDIANA LOTTO
Key: 93, Name: treo 700w
Key: 123, Name: Wrsi Riversound cafe cd
Key: 161, Name: Dillons/Kroger Employee Coupon ($1.25 credit)
Key: 178, Name: Outdoor Bag
Key: 1090, amount: VAR, Name: Rountech Asset 1
Key: 1205, amount: 1 gal, Name: GIANT NATURAL MOUNTAIN SPRING WATER
Key: 1243, Name: CVS Photo 1-Hour 4x6 Finishing
Key: 1601, Name: Sainsbury's Red Pepper
Key: 2288, amount: 18 oz, Name: Winn Dixie Hand Lotion
Build Tree Time: 5918900 nanoseconds, 5.9189 milliseconds, or 0.0059189 seconds
```

Figure 4: Insert() UPC 10

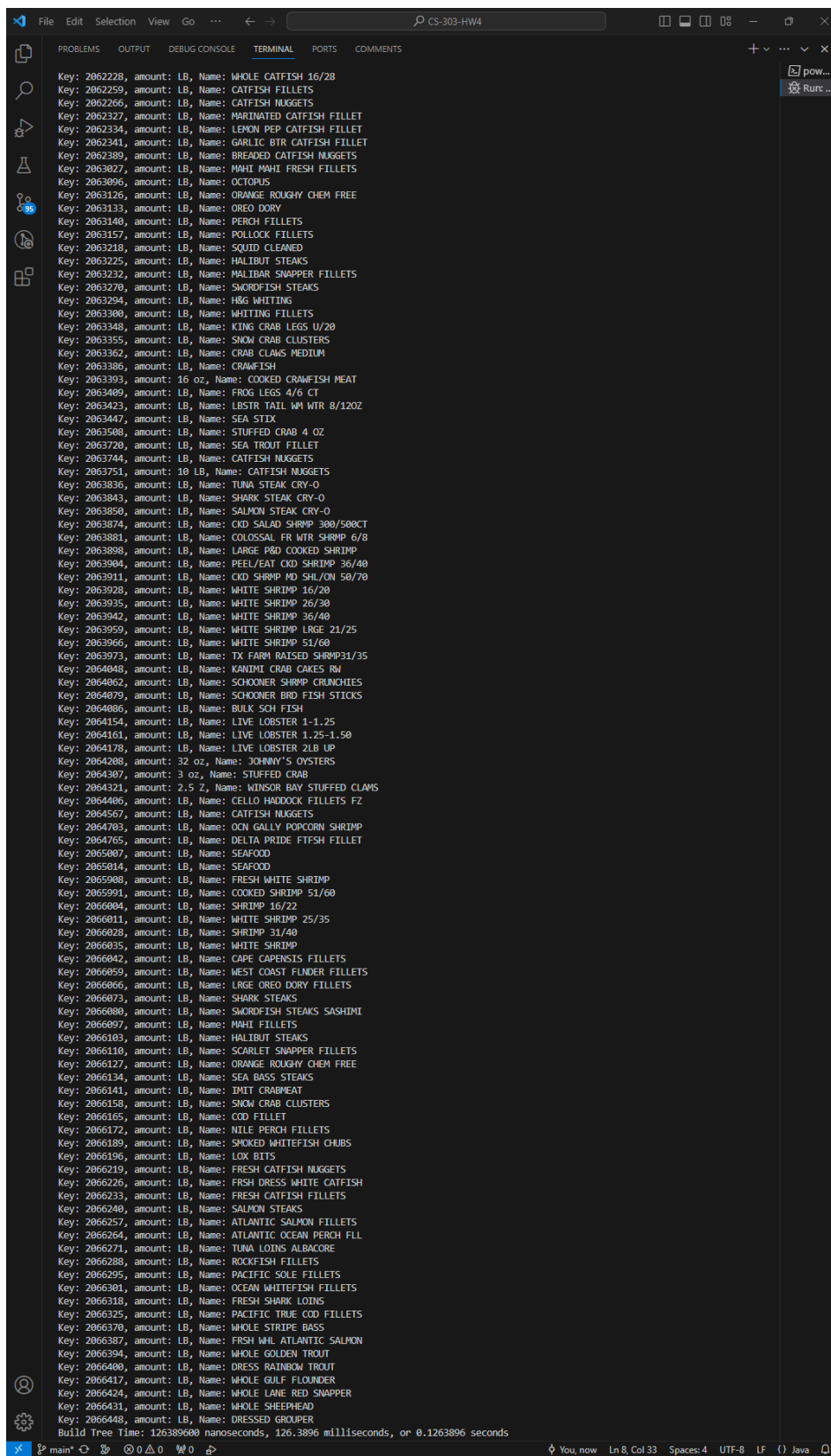


```

P5 C:\Users\logan\OneDrive - UAB - The University of Alabama at Birmingham\UAB Files\FA 2023\CS-303\HW\CS-303-HW4> c:; cd 'c:\Use
rs\logan\OneDrive - UAB - The University of Alabama at Birmingham\UAB Files\FA 2023\CS-303\HW\CS-303-HW4'; & 'C:\Program Files\Jav
a\jdk-19\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\logan\OneDrive - UAB - The University of Alabama
at Birmingham\UAB Files\FA 2023\CS-303\HW\CS-303-HW4\bin' 'HW4'
In-order traversal print:
Key: 79, Name: INDIANA LOTTO
Key: 93, Name: tree 700w
Key: 123, Name: Wsls Riversound cafe cd
Key: 161, Name: Dillons/Kroger Employee Coupon ($1.25 credit)
Key: 178, Name: Outdoor Bag
Key: 1090, amount: VAR, Name: Rountech Asset 1
Key: 1205, amount: 1 gal, Name: GIANT NATURAL MOUNTAIN SPRING WATER
Key: 1243, Name: CVS Photo 1-Hour 4x6 Finishing
Key: 1601, Name: Sainsbury's Red Pepper
Key: 2288, amount: 18 oz, Name: Winn Dixie Hand Lotion
Key: 2745, Name: sunglasses
Key: 4138, amount: 100 pack, Name: Taiyo-Yuden Value Line 8x DVD-R
Key: 4145, amount: 7.6 cm x 1.8m, Name: Conforming Bandages
Key: 5210, Name: maggi
Key: 6217, amount: 1 gal, Name: Trader Joe's 1% Lowfat Milk
Key: 7511, amount: 44 oz, Name: Maverik Plastic Soda Cup
Key: 8570, Name: jewerly
Key: 9997, amount: 32 FL OZ, Name: Down to Earth - Goji Berry Juice
Key: 10016, amount: 16.9 oz, Name: Coupon
Key: 10023, amount: 1.3 - 2.2 oz, Name: Coupon
Key: 10344, amount: CD, Name: WNCI Morning Zoo - Oops... We Did It Again!
Key: 14946, amount: 6.5mm thick - 3m roll, Name: Florist Ribbon - Lemon Yellow
Key: 29438, amount: 16 FL OZ. (1 pt) 473ML, Name: Trader Joe's Energy Drink Wild Berry
Key: 40112, amount: Variable Weight, Name: Bananas
Key: 42215, amount: Small, Name: Avacados - Green
Key: 43816, amount: 1.76 oz, Name: Altoids/ Peppermint
Key: 46268, Name: JICAMA
Key: 49887, amount: 1 x 8 oz, Name: Coca Cola:Diet Coke
Key: 50340, Name: 12057 2X72X150 POULTRY NETTING
Key: 52559, amount: 17.5 oz, Name: Raid Ant & Roach Spray
Key: 54607, Name: Engizer e2 Batteries
Key: 56731, Name: 3/8X1.5X48 POINT LATH STAKE
Key: 56748, Name: 3/8X1.5X48 WD PLASTER LATH
Key: 58322, amount: 10 oz (284g), Name: Trader Joe's Dark Chocolate Covered Macadamia Nuts
Key: 61339, Name: 4 Way Smoothing Block Nail Buffer
Key: 66419, Name: 854000 LUMBER CRAYON HOLDER
Key: 68529, Name: Gian Eagle $5 Off seafood Purchase
Key: 72342, Name: WAVELINE NON/ASBESTOS SHINGLE
Key: 72359, Name: STRT EDGE NON ASBESTOS SHINGLE
Key: 89906, amount: 16 oz (1 lb) 454g, Name: Trader Joe's Pacific Northwest Super Sweet Cut White Corn (frozen)
Key: 93699, Name: 8.ftmg.C3PwDxP6Ca.
Key: 98601, Name: 00414 9GA GALV 50# SMOOTH WIRE
Key: 98632, Name: 00426 12GA 50# GLV SMOOTH WIRE
Key: 98649, Name: 00434 14GA 50# GLV SMOOTH WIRE
Key: 98656, Name: 00442 16GA 50# GLV SMOOTH WIRE
Key: 98670, Name: 00314 9GA 50# BLAK SMOOTH WIRE
Key: 98687, Name: 00326 12GA 50# BL SMOOTH WIRE
Key: 98694, Name: 00342 16GA 50# BL SMOOTH WIRE
Key: 98724, amount: 8.5 oz (241g), Name: Trader Joe's Julianne Sliced Sun Dried Tomatoes in Olive Oil
Key: 98823, Name: 36 1X20 50 HEX NETTING
Key: 98830, Name: 48 1X20 50 HEX NETTING
Key: 99288, Name: 2X1X24X100 14GA WELD WIRE
Key: 99325, Name: 2X1X72X100 14GA WELD WIRE
Key: 99394, Name: 36X50 2X2 HDWR CLOTH
Key: 99400, Name: 48X50 2X2 HDWR CLOTH
Key: 99530, amount: 6 oz., Name: Trader Joe's Canned Crab Meat
Key: 99622, Name: 3X3 96 PAINT METAL LATH
Key: 102148, Name: CMH-4 4D MASONRY CUT NAILS
Key: 102155, Name: CMH-6 6D MASONRY CUT NAILS
Key: 102162, Name: CMH-8 8D MASONRY CUT NAILS
Key: 102179, Name: CMH-10 10D MASONRY CUT NAILS
Key: 102186, Name: CMH-16 16D MASONRY CUT NAILS
Key: 105156, Name: 19803 4-1/2 BAR TIE
Key: 105163, Name: 19804 5 BAR TIE
Key: 105170, Name: 19805 5-1/2 BAR TIE
Key: 105187, Name: 19806 6 BAR TIE
Key: 105200, Name: 19810 8 BAR TIE
Key: 114936, amount: 1 litre, Name: Indian Tonic Water
Key: 125628, amount: 454g (serves 4), Name: Sainsburys christmas pudding
Key: 129725, amount: 13.75 oz (390 g), Name: Trader Joe's Artichoke Hearts
Key: 130530, amount: 860gm, Name: Sainsbury's 48 wholewheat biscuits
Key: 140003, Name: Kroger $5.00 in Free Groceries w/ $20.00 Min. Purchase
Key: 156103, amount: 547268|0|20|CD|4|35.98, Name: CDMSIM
Key: 157148, amount: 750 mL, Name: DOM LEFLAIVE BATARD MONTRACHET BIENVENUES
Key: 165105, amount: 16 oz., Name: Trader Joe's Mini Ravioli With Cheese Filling
Key: 179744, amount: 16 oz (1 lb) 454g, Name: Trader Joe's Marinated Mushrooms with Garlic
Key: 188920, amount: 1 lb. 8oz. (681g), Name: TRADER JOE'S WHITE CREAMER POTATOES SKU#18892
Key: 200318, amount: 16 oz., Name: Trader Joe's Green Protein
Key: 210805, amount: Net Wt. 2LB 8 oz (1135g), Name: TRADER JOE'S RUSSET BAKER POTATOES SKU#21080
Key: 235525, amount: 5.5 Oz., Name: ""Trader Joe's"" Tuna for Cats
Key: 259941, amount: 25.3 FL OZ, Name: Trader Joes Natural Mountain Spring Water
Key: 267304, amount: 280g, Name: Marks & Spencer Teacakes (16) Milk Chocolate
Key: 270427, amount: 907g, Name: organic sugar
Key: 298803, Name: 7 7X1/4 FLOAT BALL
Key: 310222, Name: 1/2X21 T&C BLACK IMP PIPE
Key: 310239, Name: 3/4X21 T&C BLK IMP PIPE
Key: 310246, Name: 1X21 T&C BLACK IMP PIPE
Key: 310253, Name: 1-1/4X21 T&C BLACK IMP PIPE
Key: 310260, Name: 1-1/2X21 T&C BLACK IMP PIPE
Key: 310277, Name: 2X21 T&C BLACK IMP PIPE
Key: 313452, Name: 1/2X21 T&C GALV IMP PIPE
Key: 313469, Name: 3/4X21 T&C GALV IMP PIPE
Key: 313476, Name: 1X21 T&C GALV IMP PIPE
Build Tree Time: 25126900 nanoseconds, 25.1269 milliseconds, or 0.0251269 seconds

```

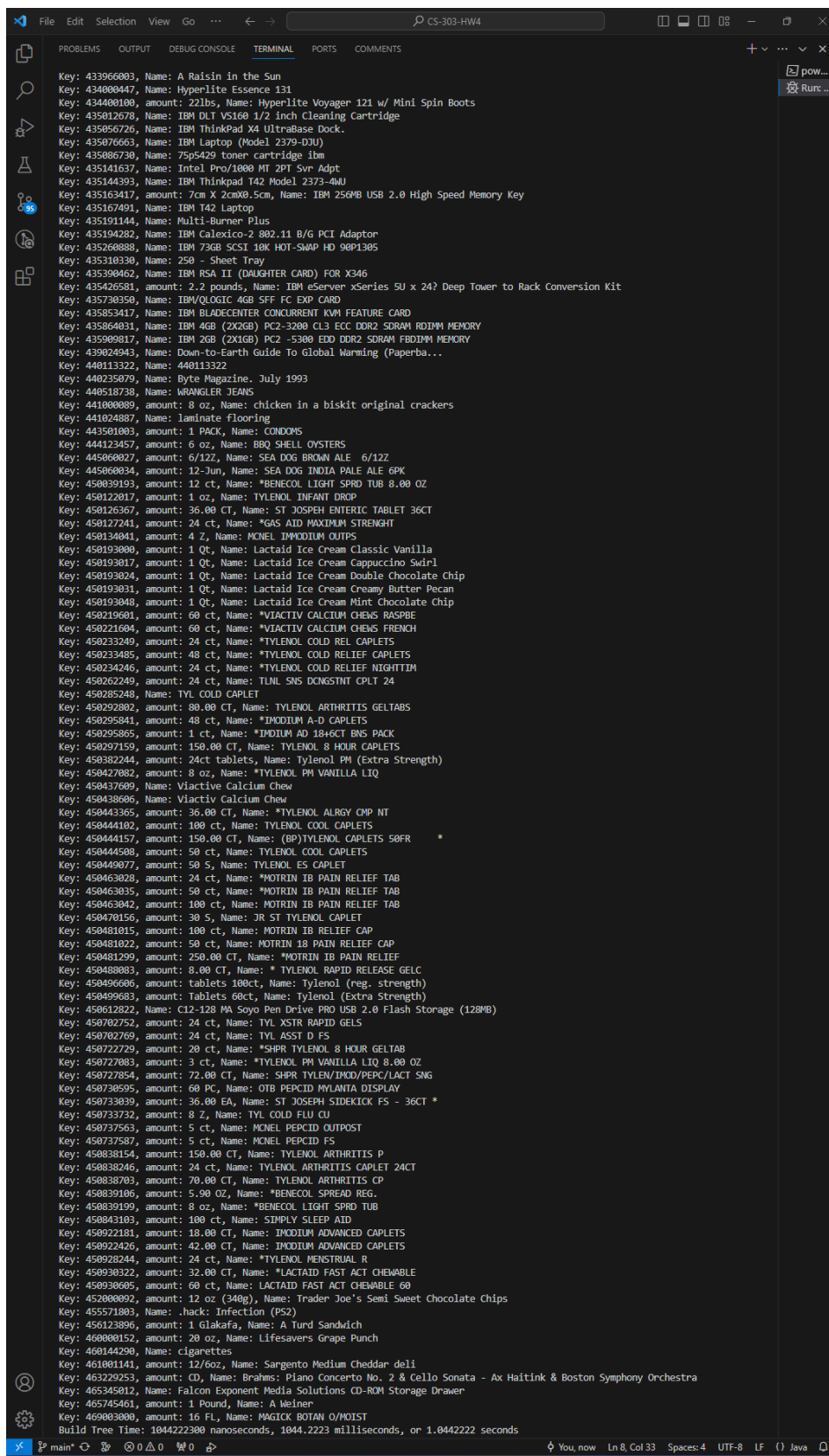
Figure 5: Insert() UPC 100



The image shows a screenshot of an IDE's terminal window. The terminal displays a list of UPC keys and their corresponding item names. The list starts with 'Key: 2062228, amount: LB, Name: WHOLE CATFISH 16/28' and continues down to 'Key: 2066448, amount: LB, Name: DRESSED GROUPER'. The terminal window has a dark theme and includes a sidebar with icons for file explorer, search, and other IDE features. The status bar at the bottom indicates the current file is 'main', the cursor is at line 8, column 33, and the encoding is UTF-8.

```
File Edit Selection View Go ... CS-303-HW4
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Key: 2062228, amount: LB, Name: WHOLE CATFISH 16/28
Key: 2062259, amount: LB, Name: CATFISH FILLETS
Key: 2062266, amount: LB, Name: CATFISH MUGGETS
Key: 2062327, amount: LB, Name: MARINATED CATFISH FILLET
Key: 2062334, amount: LB, Name: LEMON PEP CATFISH FILLET
Key: 2062341, amount: LB, Name: GARLIC BTR CATFISH FILLET
Key: 2062389, amount: LB, Name: BREADED CATFISH MUGGETS
Key: 2063027, amount: LB, Name: MAHI MAHI FRESH FILLETS
Key: 2063096, amount: LB, Name: OCTOPUS
Key: 2063126, amount: LB, Name: ORANGE ROUGHY CHEM FREE
Key: 2063133, amount: LB, Name: OREG DORY
Key: 2063140, amount: LB, Name: PERCH FILLETS
Key: 2063157, amount: LB, Name: POLLOCK FILLETS
Key: 2063210, amount: LB, Name: SQUID CLEANED
Key: 2063225, amount: LB, Name: HALIBUT STEAKS
Key: 2063232, amount: LB, Name: MALIBAR SNAPPER FILLETS
Key: 2063270, amount: LB, Name: SMORDFISH STEAKS
Key: 2063294, amount: LB, Name: H&G WHITING
Key: 2063300, amount: LB, Name: WHITING FILLETS
Key: 2063348, amount: LB, Name: KING CRAB LEGS U/20
Key: 2063355, amount: LB, Name: SNOW CRAB CLUSTERS
Key: 2063362, amount: LB, Name: CRAB CLAMS MEDIUM
Key: 2063386, amount: LB, Name: CRAWFISH
Key: 2063393, amount: 16 oz, Name: COOKED CRAWFISH MEAT
Key: 2063409, amount: LB, Name: FROG LEGS 4/6 CT
Key: 2063423, amount: LB, Name: LBSTR TAIL WH WTR 8/12OZ
Key: 2063447, amount: LB, Name: SEA STIX
Key: 2063508, amount: LB, Name: STUFFED CRAB 4 OZ
Key: 2063720, amount: LB, Name: SEA TROUT FILLET
Key: 2063744, amount: LB, Name: CATFISH MUGGETS
Key: 2063751, amount: 10 LB, Name: CATFISH MUGGETS
Key: 2063836, amount: LB, Name: TUNA STEAK CRY-O
Key: 2063843, amount: LB, Name: SHARK STEAK CRY-O
Key: 2063850, amount: LB, Name: SALMON STEAK CRY-O
Key: 2063874, amount: LB, Name: CKD SALAD SHRMP 300/500CT
Key: 2063881, amount: LB, Name: COLOSSAL FR WTR SHRMP 6/8
Key: 2063898, amount: LB, Name: LARGE P&D COOKED SHRIMP
Key: 2063904, amount: LB, Name: PEEL/EAT CKD SHRIMP 36/40
Key: 2063911, amount: LB, Name: CKD SHRMP MD SHL/ON 50/70
Key: 2063928, amount: LB, Name: WHITE SHRIMP 16/20
Key: 2063935, amount: LB, Name: WHITE SHRIMP 26/30
Key: 2063942, amount: LB, Name: WHITE SHRIMP 36/40
Key: 2063959, amount: LB, Name: WHITE SHRIMP LGCE 21/25
Key: 2063966, amount: LB, Name: WHITE SHRIMP 51/60
Key: 2063973, amount: LB, Name: TX FARM RAISED SHRMP31/35
Key: 2064048, amount: LB, Name: KANIMI CRAB CAKES RM
Key: 2064062, amount: LB, Name: SCHONER SHRMP CRUNCHIES
Key: 2064079, amount: LB, Name: SCHONER BRD FISH STICKS
Key: 2064086, amount: LB, Name: BULK SCH FISH
Key: 2064154, amount: LB, Name: LIVE LOBSTER 1-1.25
Key: 2064161, amount: LB, Name: LIVE LOBSTER 1.25-1.50
Key: 2064178, amount: LB, Name: LIVE LOBSTER 2LB UP
Key: 2064208, amount: 32 oz, Name: JOHNNY'S OYSTERS
Key: 2064307, amount: 3 oz, Name: STUFFED CRAB
Key: 2064321, amount: 2.5 Z, Name: WINSOR BAY STUFFED CLAMS
Key: 2064406, amount: LB, Name: CELLO HADDOCK FILLETS FZ
Key: 2064567, amount: LB, Name: CATFISH MUGGETS
Key: 2064703, amount: LB, Name: OCH GALLY POPCORN SHRIMP
Key: 2064765, amount: LB, Name: DELTA PRIDE FTFSH FILLET
Key: 2065007, amount: LB, Name: SEAFOOD
Key: 2065014, amount: LB, Name: SEAFOOD
Key: 2065080, amount: LB, Name: FRESH WHITE SHRIMP
Key: 2065091, amount: LB, Name: COOKED SHRIMP 51/60
Key: 2066004, amount: LB, Name: SHRIMP 16/22
Key: 2066011, amount: LB, Name: WHITE SHRIMP 25/35
Key: 2066028, amount: LB, Name: SHRIMP 31/40
Key: 2066035, amount: LB, Name: WHITE SHRIMP
Key: 2066042, amount: LB, Name: CAPE CAPENSIS FILLETS
Key: 2066059, amount: LB, Name: WEST COAST FLUNDER FILLETS
Key: 2066066, amount: LB, Name: LRGE OREG DORY FILLETS
Key: 2066073, amount: LB, Name: SHARK STEAKS
Key: 2066080, amount: LB, Name: SMORDFISH STEAKS SASHIMI
Key: 2066097, amount: LB, Name: MAHI FILLETS
Key: 2066103, amount: LB, Name: HALIBUT STEAKS
Key: 2066110, amount: LB, Name: SCARLET SNAPPER FILLETS
Key: 2066127, amount: LB, Name: ORANGE ROUGHY CHEM FREE
Key: 2066134, amount: LB, Name: SEA BASS STEAKS
Key: 2066141, amount: LB, Name: IMIT CRABMEAT
Key: 2066150, amount: LB, Name: SNOW CRAB CLUSTERS
Key: 2066165, amount: LB, Name: COD FILLET
Key: 2066172, amount: LB, Name: NILE PERCH FILLETS
Key: 2066189, amount: LB, Name: SMOKED WHITEFISH CHUBS
Key: 2066196, amount: LB, Name: LOX BITS
Key: 2066219, amount: LB, Name: FRESH CATFISH MUGGETS
Key: 2066226, amount: LB, Name: FRSH DRESS WHITE CATFISH
Key: 2066233, amount: LB, Name: FRESH CATFISH FILLETS
Key: 2066240, amount: LB, Name: SALMON STEAKS
Key: 2066257, amount: LB, Name: ATLANTIC SALMON FILLETS
Key: 2066264, amount: LB, Name: ATLANTIC OCEAN PERCH FLL
Key: 2066271, amount: LB, Name: TUNA LOINS ALBACORE
Key: 2066288, amount: LB, Name: ROCKFISH FILLETS
Key: 2066295, amount: LB, Name: PACIFIC SOLE FILLETS
Key: 2066301, amount: LB, Name: OCEAN WHITEFISH FILLETS
Key: 2066318, amount: LB, Name: FRESH SHARK LOINS
Key: 2066325, amount: LB, Name: PACIFIC TRUE COD FILLETS
Key: 2066370, amount: LB, Name: WHOLE STRIPE BASS
Key: 2066387, amount: LB, Name: FRSH WHL ATLANTIC SALMON
Key: 2066394, amount: LB, Name: WHOLE GOLDEN TROUT
Key: 2066400, amount: LB, Name: DRESS RAINBOW TROUT
Key: 2066417, amount: LB, Name: WHOLE GULF FLOUNDER
Key: 2066424, amount: LB, Name: WHOLE LANE RED SNAPPER
Key: 2066431, amount: LB, Name: WHOLE SHEEPHEAD
Key: 2066448, amount: LB, Name: DRESSED GROUPER
Build Tree Time: 126389600 nanoseconds, 126.3896 milliseconds, or 0.1263896 seconds
You, now Ln 8, Col 33 Spaces: 4 UTF-8 LF Java
```

Figure 6: Insert() UPC 1K



```
File Edit Selection View Go ... CS-303-HW4
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Key: 433966003, Name: A Raisin in the Sun
Key: 434000447, Name: Hyperlite Essence 131
Key: 434400100, amount: 22lbs, Name: Hyperlite Voyager 121 w/ Mini Spin Boots
Key: 435012678, Name: IBM DLT V5160 1/2 inch Cleaning Cartridge
Key: 435056726, Name: IBM ThinkPad X4 UltraBase Dock.
Key: 435076663, Name: IBM Laptop (Model 2379-D3U)
Key: 435086730, Name: 75p5429 toner cartridge ibm
Key: 435141637, Name: Intel Pro/1000 MT 2PT Svr Adpt
Key: 435144393, Name: IBM Thinkpad T42 Model 2373-4WJ
Key: 435163417, amount: 7cm X 2cmX0.5cm, Name: IBM 256MB USB 2.0 High Speed Memory Key
Key: 435167491, Name: IBM T42 Laptop
Key: 435191144, Name: Multi-Burner Plus
Key: 435194202, Name: IBM Galaxico-2 802.11 B/G PCI Adaptor
Key: 435260880, Name: IBM 73GB SCSI 10K HOT-SWAP HD 90P1305
Key: 435310330, Name: 250 - Sheet Tray
Key: 435309462, Name: IBM RSA II (DAUGHTER CARD) FOR X346
Key: 435426581, amount: 2.2 pounds, Name: IBM eServer xSeries 5U x 24? Deep Tower to Rack Conversion Kit
Key: 435720350, Name: IBM/LOGIC 4GB SFF FC EXP CARD
Key: 435853417, Name: IBM BLADECENTER CONCURRENT KVM FEATURE CARD
Key: 435864031, Name: IBM 4GB (2X2GB) PC2-3200 CL3 ECC DDR2 SDRAM RDRAM MEMORY
Key: 435909017, Name: IBM 2GB (2X1GB) PC2 -5300 EDD DDR2 SDRAM FBDRAM MEMORY
Key: 439024943, Name: Down-to-Earth Guide To Global Warming (Paperba...
Key: 440113322, Name: 440113322
Key: 440235079, Name: Byte Magazine. July 1993
Key: 440518738, Name: WRANGLER JEANS
Key: 441000089, amount: 8 oz, Name: chicken in a biskit original crackers
Key: 441024887, Name: laminate flooring
Key: 443501003, amount: 1 PACK, Name: CORDONS
Key: 444123457, amount: 6 oz, Name: BBQ SHELL OYSTERS
Key: 445060027, amount: 6/12Z, Name: SEA DOG BROWN ALE 6/12Z
Key: 445060034, amount: 12-3un, Name: SEA DOG INDIA PALE ALE 6PK
Key: 450039193, amount: 12 ct, Name: *BENECOL LIGHT SPRD TUB 8.00 OZ
Key: 450122017, amount: 1 oz, Name: TYLENOL INFANT DROP
Key: 450126367, amount: 36.00 CT, Name: ST JOSEPH ENTERIC TABLET 36CT
Key: 450127241, amount: 24 ct, Name: *GAS AID MAXIMUM STRENGHT
Key: 450134041, amount: 4 Z, Name: MCNEL IMMIDIUM OUTPS
Key: 450193000, amount: 1 Qt, Name: Lactaid Ice Cream Classic Vanilla
Key: 450193017, amount: 1 Qt, Name: Lactaid Ice Cream Cappuccino Swirl
Key: 450193024, amount: 1 Qt, Name: Lactaid Ice Cream Double Chocolate Chip
Key: 450193031, amount: 1 Qt, Name: Lactaid Ice Cream Creamy Butter Pecan
Key: 450193048, amount: 1 Qt, Name: Lactaid Ice Cream Mint Chocolate Chip
Key: 450219501, amount: 60 ct, Name: *VIACTIVE CALCIUM CHEWS RASPB
Key: 450221604, amount: 60 ct, Name: *VIACTIVE CALCIUM CHEWS FRENCH
Key: 450233249, amount: 24 ct, Name: *TYLENOL COLD REL CAPLETS
Key: 450233485, amount: 48 ct, Name: *TYLENOL COLD RELIEF CAPLETS
Key: 450234246, amount: 24 ct, Name: *TYLENOL COLD RELIEF NIGHTTIM
Key: 450262249, amount: 24 ct, Name: TULIL SNS DONGSTHT CPLT 24
Key: 450285248, Name: TYL COLD CAPLET
Key: 450292802, amount: 80.00 CT, Name: TYLENOL ARTHRITIS GELTABS
Key: 450295041, amount: 48 ct, Name: *IMODIUM A-D CAPLETS
Key: 450295065, amount: 1 ct, Name: *IMODIUM AD 18+6CT BNS PACK
Key: 450297159, amount: 150.00 CT, Name: TYLENOL 8 HOUR CAPLETS
Key: 450382244, amount: 24ct tablets, Name: Tylenol PM (Extra Strength)
Key: 450427082, amount: 8 oz, Name: *TYLENOL PM VANILLA LIQ
Key: 450437609, Name: Viactive Calcium Chew
Key: 450438606, Name: Viactiv Calcium Chew
Key: 450443365, amount: 36.00 CT, Name: *TYLENOL ALRGY CMP NT
Key: 450444102, amount: 100 ct, Name: TYLENOL COOL CAPLETS
Key: 450444157, amount: 150.00 CT, Name: (80)TYLENOL CAPLETS 50FR *
Key: 450444508, amount: 50 ct, Name: TYLENOL COOL CAPLETS
Key: 450449077, amount: 50 S, Name: TYLENOL ES CAPLET
Key: 450463028, amount: 24 ct, Name: *MOTRIN IB PAIN RELIEF TAB
Key: 450463035, amount: 50 ct, Name: *MOTRIN IB PAIN RELIEF TAB
Key: 450463042, amount: 100 ct, Name: MOTRIN IB PAIN RELIEF TAB
Key: 450470156, amount: 30 S, Name: JR ST TYLENOL CAPLET
Key: 450481015, amount: 100 ct, Name: MOTRIN IB RELIEF CAP
Key: 450481022, amount: 50 ct, Name: MOTRIN 18 PAIN RELIEF CAP
Key: 450481299, amount: 250.00 CT, Name: *MOTRIN IB PAIN RELIEF
Key: 450480083, amount: 8.00 CT, Name: * TYLENOL RAPID RELEASE GELC
Key: 450496606, amount: tablets 100ct, Name: Tylenol (reg. strength)
Key: 450496683, amount: Tablets 60ct, Name: Tylenol (Extra Strength)
Key: 450612822, Name: C12-128 MA Soyo Pen Drive PRO USB 2.0 Flash Storage (128MB)
Key: 450702752, amount: 24 ct, Name: TYL XSTR RAPID GELS
Key: 450702769, amount: 24 ct, Name: TYL ASST D FS
Key: 450722729, amount: 20 ct, Name: *SHPR TYLENOL 8 HOUR GELTAB
Key: 450727083, amount: 3 ct, Name: *TYLENOL PM VANILLA LIQ 8.00 OZ
Key: 450727854, amount: 72.00 CT, Name: SHPR TYLEN/IMOD/PEPC/LACT SNG
Key: 450730595, amount: 60 PC, Name: OTB PEPCID MYLANITA DISPLAY
Key: 450733039, amount: 36.00 EA, Name: ST JOSEPH SIDEKICK FS - 36CT *
Key: 450733732, amount: 8 Z, Name: TYL COLD FLU CU
Key: 450737563, amount: 5 ct, Name: MCNEL PEPCID OUTPOST
Key: 450737587, amount: 5 ct, Name: MCNEL PEPCID FS
Key: 450838154, amount: 150.00 CT, Name: TYLENOL ARTHRITIS P
Key: 450838246, amount: 24 ct, Name: TYLENOL ARTHRITIS CAPLET 24CT
Key: 450838703, amount: 70.00 CT, Name: TYLENOL ARTHRITIS CP
Key: 450839106, amount: 5.90 OZ, Name: *BENECOL SPREAD REG.
Key: 450839199, amount: 8 oz, Name: *BENECOL LIGHT SPRD TUB
Key: 450843103, amount: 100 ct, Name: SIMPLY SLEEP AID
Key: 450922101, amount: 18.00 CT, Name: IMODIUM ADVANCED CAPLETS
Key: 450922426, amount: 42.00 CT, Name: IMODIUM ADVANCED CAPLETS
Key: 450920244, amount: 24 ct, Name: *TYLENOL MENSTRUAL R
Key: 450930322, amount: 32.00 CT, Name: *LACTAID FAST ACT CHEWABLE
Key: 450930605, amount: 60 ct, Name: LACTAID FAST ACT CHEWABLE 60
Key: 452000092, amount: 12 oz (340g), Name: Trader Joe's Semi Sweet Chocolate Chips
Key: 455571803, Name: .hack: Infection (PS2)
Key: 456123096, amount: 1 Glakafa, Name: A Turd Sandwich
Key: 460000152, amount: 20 oz, Name: Lifesavers Grape Punch
Key: 460144290, Name: cigarettes
Key: 461001141, amount: 12/6oz, Name: Sargento Medium Cheddar dell
Key: 463229253, amount: CD, Name: Brahms: Piano Concerto No. 2 & Cello Sonata - Ax Haitink & Boston Symphony Orchestra
Key: 465345012, Name: Falcon Exponent Media Solutions CD-ROM Storage Drawer
Key: 465745461, amount: 1 Pound, Name: A Weiner
Key: 469003000, amount: 16 FL, Name: MAGIC BOTAN O/MOIST
Build Tree Time: 1044222300 nanoseconds, 1044.2223 milliseconds, or 1.0442222 seconds
X You, now Ln 8, Col 33 Spaces: 4 UTF-8 LF ( ) Java Q
```

Figure 7: Insert() UPC 10K

The image shows a screenshot of an IDE window with the title bar "CS-303-HW4". The interface includes a menu bar (File, Edit, Selection, View, Go, ...), a toolbar with icons for file operations, and a sidebar with icons for Explorer, Search, Run and Debug, and Extensions. The main area is divided into four panes: PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The TERMINAL pane is active and displays a long list of items, each with a key, amount, and name. The items are listed in a single column, with some items having an asterisk (\*) next to them. The status bar at the bottom shows "main", "You, 1 second ago", "Ln 8, Col 34", "Spaces: 4", "UTF-8", "LF", and "Java".

```
Key: 18894316725, amount: 7 oz, Name: BIG Y CHED & BCON POT SKINS
Key: 18894316732, amount: 8 oz, Name: *BIG Y CHDR CHSE JALEPO POPP
Key: 18894316749, amount: 8 oz, Name: BIG Y MOZZRELLA STICKS
Key: 18894316756, amount: 8 oz, Name: BY PORK MINI EGG ROLLS FRZ
Key: 18894318002, amount: 80.00 OZ, Name: BY WH BRD DO 5 LVS 31800
Key: 18894318019, amount: 16 oz, Name: BY WHT DIN ROL 31801
Key: 18894318026, amount: 8 oz, Name: FZ GARLIC DINNER ROLLS 6CT *
Key: 18894319816, Name: BIG Y DOZEN LARGE WHITE EGGS
Key: 18894319696, amount: 10 oz, Name: BY BROCCOLI W/CHEESE SAUCE
Key: 18894319702, amount: 10 oz, Name: BY BROCCOLI CARR W/CHEESE
Key: 18894319719, amount: 10 oz, Name: BY SWEET CORN & BUTTER SAUCE
Key: 18894319726, amount: 9 oz, Name: BY CREAMED SPINACH
Key: 18894319733, amount: 16 oz, Name: BY EDAMAME W/PODS
Key: 18894319740, amount: 16 oz, Name: BY SLICED YELLOW SQUASH
Key: 18894319757, amount: 16 oz, Name: BY SLICED ZUCCHINI
Key: 18894319764, amount: 12 oz, Name: BY DICED ONIONS
Key: 18894319771, amount: 16 oz, Name: BY CUT LEAF SPINACH
Key: 18894319788, amount: 21.00 OZ, Name: *BY MEAL STRIR TERYAKI STIR F
Key: 18894319818, amount: 21.00 OZ, Name: *BY MEAL STRIR LO MEIN STR FR
Key: 18894319825, amount: 10 oz, Name: BY PEAS AND CARROTS
Key: 18894319832, amount: 12 oz, Name: BY CK WINTER SQUASH
Key: 18894319849, amount: 24 ct, Name: BY CORN COB-24 MINI EARS
Key: 18894319856, amount: 12 oz, Name: BY SUGAR SNAP PEAS
Key: 18894319863, amount: 12 oz, Name: BY PETITE WHOLE KERNEL CORN
Key: 18894319870, amount: 16 oz, Name: BY PEAS & CARROTS
Key: 18894319887, amount: 10 oz, Name: BY MIXED VEGGIES
Key: 18894319894, amount: 9 oz, Name: BY GREEN BEANS ITALIAN TYPE
Key: 18894319900, amount: 4 ct, Name: BY CORN ON COB 4 EARS
Key: 18894319917, amount: 16 oz, Name: BY CAULIFLOWER FLORETS
Key: 18894319931, amount: 16 oz, Name: BY CRINKLE CUT CARROTS
Key: 18894319948, amount: 16 oz, Name: BY ONION RINGS
Key: 18894319955, amount: 16 oz, Name: BY BRUSSELS SPROUTS
Key: 18894319962, amount: 10 oz, Name: BY BRUSSELS SPROUTS
Key: 18894319986, amount: 10 oz, Name: BY BABY LIMA BEANS
Key: 18894319993, amount: 16 oz, Name: BY BROCCOLI SPEARS
Key: 18894320005, amount: 16 oz, Name: BY CORN 32000
Key: 18894320012, amount: 16 oz, Name: BY CUT GREEN BEAN 32001
Key: 18894320029, amount: 16 oz, Name: BY GREEN PEA 32002
Key: 18894320036, amount: 16 oz, Name: BY MIXED VEG 32003
Key: 18894320043, amount: 16 oz, Name: BY WHL BABY CARROTS 32004
Key: 18894320050, amount: 16 oz, Name: BY BROCCOLI CUT 32005
Key: 18894320067, amount: 16 oz, Name: BY FLORENTINE BLND 32006
Key: 18894320074, amount: 16 oz, Name: BY CALIF BLND 32007
Key: 18894320081, amount: 16 oz, Name: BY GREEN BEANS FRENCH CUT
Key: 18894320098, amount: 16 oz, Name: BY PETITE GREEN PEAS
Key: 18894320104, amount: 16 oz, Name: BY ORIENTAL STYLE VEGGS
Key: 18894320111, amount: 16 oz, Name: BY ITALIAN STYLE VEGETABLES
Key: 18894320128, amount: 32 oz, Name: BY BROCCOLI CUTS
Key: 18894320135, amount: 32 oz, Name: BY CUT WHOLE KERNAL CORN
Key: 18894320142, amount: 32 oz, Name: BY GREEN PEAS
Key: 18894320159, amount: 10 oz, Name: BY BROCCOLI SPEARS
Key: 18894320166, amount: 10 oz, Name: BY BROCCOLI CHOPPED
Key: 18894320173, amount: 10 oz, Name: BY CUT GOLDEN CORN
Key: 18894320180, amount: 10 oz, Name: BY GREEN PEA 32018
Key: 18894320197, amount: 10 oz, Name: BY SPINACH CHOPPED
Key: 18894320203, amount: 10 oz, Name: BY SPINACH LEAF
Key: 18894320210, amount: 32 oz, Name: BY MIXED VEGETABLE
Key: 18894320227, amount: 16 oz, Name: BY MEXICAN STYLE VEGGIES
Key: 18894320234, amount: 9 oz, Name: BY CUT GREEN BEANS
Key: 18894320259, amount: 10 oz, Name: BY FORDHOOK LIMA BEANS
Key: 18894320272, amount: 8 oz, Name: BY WHIPPED TOPPING
Key: 18894320296, amount: 12 oz, Name: BY WHIPPED TOPPING
Key: 18894320302, amount: 16 oz, Name: BY SLICED STRAWBERRIES
Key: 18894320319, amount: 16 oz, Name: BY WHOLE STRAWBERRIES
Key: 18894320333, amount: 15 oz, Name: BY LITE SLICED STRAWBERRIES
Key: 18894320340, amount: 16 oz, Name: BY BLUEBERRIES UNSMT
Key: 18894320364, amount: 16 oz, Name: BY MIXED FRUIT UNSMT
Key: 18894320371, amount: 12 oz, Name: BY RED RASPBERRIES
Key: 18894320388, amount: 12 oz, Name: BY BERRY MEDLEY
Key: 18894320395, amount: 8 oz, Name: BY FF WHIPPED TOPPING
Key: 18894320418, amount: 64 oz, Name: BY WHOLE STRAWBERRIES
Key: 18894320449, amount: 16 oz, Name: BY TROPICAL FRUIT BLEND
Key: 18894320487, amount: 16 oz, Name: BY VEGETABLES-STEW MIX
Key: 18894320494, amount: 16 oz, Name: BY VEGETABLES-STIR FRY
Key: 18894321545, amount: 12 oz, Name: BY GRAPE JELLY
Key: 18894323624, amount: 18 oz, Name: BY GRAPE JELLY
Key: 18894323648, amount: 32.00 FO, Name: BY GRAPE JELLY
Key: 18894324966, amount: 28 oz, Name: BIG Y CRISPY CURLY CUT FRIES
Key: 18894324973, amount: 28 oz, Name: BIG Y STEAK FRIES
Key: 18894324980, amount: 28 oz, Name: BIG Y SHOESTRING POTATOES
Key: 18894325017, amount: 32 oz, Name: BY TATER TREATS
Key: 18894325024, amount: 32 oz, Name: BY CRINKLE CUT FRIES
Key: 18894325031, amount: 32 oz, Name: BY REGULAR CUT FRIES
Key: 18894325062, amount: 12.30 OZ, Name: BIG Y BLUEBERRY WAFFLES 10 CT
Key: 18894325338, amount: 12.30 OZ, Name: BIG Y OLD FASH WAFFLES 10 CT *
Key: 18894325345, amount: 12.30 OZ, Name: BIG Y BITTLMILK WAFFLES
Key: 18894325352, amount: 19.80 OZ, Name: BIG Y HOMESTYLE WAFFLES 16 CT
Key: 18894325369, amount: 19.80 OZ, Name: BIG Y BUTTERMILK WAFFLES 16 C
Key: 18894325376, amount: 19.80 OZ, Name: BIG Y BLUEBERRY WAFFLES 16 CT
Key: 18894325383, amount: 12.30 OZ, Name: BY APPLE CINAMON WAFFLES 10CT
Key: 18894328001, amount: 16 oz, Name: BY BEEF FRANK 1# 685
Key: 18894328019, amount: 16 oz, Name: BY MEAT FRANKS 1#
Key: 18894328025, amount: 16 oz, Name: BY REGULAR BACON
Key: 18894328247, amount: 48 oz, Name: BIG Y FROZEN BEEF BURGERS
Key: 18894328353, amount: 8 oz, Name: BY SLCD BOLOGNA PP 4880
Key: 18894328384, amount: 8 oz, Name: BY COOKED SALAMI PP 4884
Key: 18894328391, amount: 8 oz, Name: BY PICKLE LOAF PP 4883
Key: 18894328421, amount: 5 lb, Name: BY IQF PARTY WINGS *
Key: 18894328438, amount: 3.00 LB, Name: BY IQF BNLS SKNLS BREAST BY
Key: 18894328490, amount: 5 lb, Name: BY BEEF PATTIES 5# FAM SIZE
Key: 18894328506, amount: 8 oz, Name: BIG Y CRHY BRD FISH STX 12CT
Key: 18894328513, amount: 12 oz, Name: BIG Y CRHY BRD FISH STX 18CT
Key: 18894328537, amount: 8 oz, Name: BIG Y CRHY BRD HADDOCK FLT
Build Tree Time: 8749228600 nanoseconds, 8749.221 milliseconds, or 8.749221 seconds
```

Figure 8: Insert() UPC 100K

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS	COMMENTS
Key: 28785207452,	amount: 1 ct,	Name: VICKS HUMIDIFIER WRM,MIST			
Key: 28785255200,	Name: *KAZ PERSNAL MIST HUMIDFY				
Key: 28785300405,	Name: KAZ INHALANT				
Key: 28785300603,	amount: 6 oz,	Name: KAZ INHALANT KFC6			
Key: 28785313146,	Name: KAZ DYNAFILTER				
Key: 28785313457,	amount: 1 EA,	Name: PROTEC TANK CLEANING CARTRIDGE			
Key: 28785340020,	amount: EA,	Name: VICKS STM INHALE V1200			
Key: 28785509110,	amount: 1 ct,	Name: *VICKS THERMOMETER SPEED V912			
Key: 28785509327,	amount: 1 ct,	Name: * VICKS UNDERARM THERMOMETER			
Key: 28785509778,	amount: 1 ct,	Name: VICKS THERMOMETER FOREHEAD			
Key: 28785601106,	amount: 1 ct,	Name: KAZ HEAT PAD DRYHP 110			
Key: 28785604404,	Name: HEAT PAD				
Key: 28785607009,	Name: HEAT PAD				
Key: 28785607100,	amount: HP710,	Name: MOIST AND DRY HEAT PAD			
Key: 28785607504,	Name: HEATING PAD				
Key: 28785609102,	Name: KAZ HEATING PAD				
Key: 28785609508,	amount: "12"" X 24""",	Name: "Heating Pad ""Moist or Dry"""			
Key: 28785613024,	amount: EA,	Name: KAZ THERMI PAQ9.5X16 HC1302			
Key: 28800000013,	amount: 16 oz,	Name: Hanover Three Bean Salad Premium			
Key: 28800000037,	amount: 16 oz,	Name: HANOVER VEGETABLE SALAD			
Key: 28800000129,	amount: 15.50 OZ,	Name: HANOVER 4 BEAN SALAD			
Key: 28800000044,	amount: 7.5oz (213g),	Name: Phillips Beans & Franks			
Key: 28800000048,	amount: 7.5 OZ,	Name: HANOVER BEANS & FRANKS			
Key: 28800000050,	amount: 7.50 OZ,	Name: HANOVER KID BEANS F20			
Key: 288000000570,	amount: 7.75 OZ,	Name: Hanover Peas Chick			
Key: 288000000990,	amount: 4 ct,	Name: HANOVER SMT WHT CORN 4CT			
Key: 28800100300,	amount: 112oz (7LB),	Name: Hanover Ready-to-Serve Chocolate Pudding			
Key: 28800100508,	amount: 6 lb 9 oz,	Name: Hanover Baby Lima Beans			
Key: 28800103127,	amount: 101 oz,	Name: WC HANOVER CUT GREEN BEAN			
Key: 28800103134,	amount: 38 OZ,	Name: Hanover Beans Green Blue Lake Cut			
Key: 28800103141,	amount: 50 OZ,	Name: WC HANOVER CUT GREEN BEAN			
Key: 28800103172,	amount: 24.50 OZ,	Name: HANOVER GRN BN BLU LAKE CT			
Key: 28800109013,	amount: 39 Oz,	Name: Hanover Beans Green & Potatoes In Ham Flavored Sauce			
Key: 28800109037,	amount: 25.00 OZ,	Name: *HANOVER CT GRN BEANS &MHL PO			
Key: 28800123026,	amount: 106 oz,	Name: WC HANOVER GOLDEN CORN			
Key: 28800126119,	amount: 15 oz,	Name: HANOVER SLICED POTATOES			
Key: 28800126461,	amount: 108 oz,	Name: Hanover Small whole White Potatoes			
Key: 28800126500,	amount: 109 OZ,	Name: ALLIANT WHOLE POTATOES			
Key: 28800126812,	amount: 15 oz,	Name: HANOVER SM WHOLE POTATOES			
Key: 28800127049,	amount: 15 oz,	Name: Hanover Pasta Wheelies In Tomato Sauce			
Key: 28800127062,	amount: 19.5 OZ,	Name: HANOVER WHEELIES BONUS			
Key: 28800127208,	amount: 15 oz,	Name: HANOVER GO-GO S T/S WCHSE			
Key: 28800127222,	amount: 19.5 OZ,	Name: HANOVER GO GO S BONUS			
Key: 28800127307,	amount: 15 oz,	Name: HANOVER HOT NOTES T/S CHS			
Key: 28800127321,	amount: 19.5 OZ,	Name: HANOVER HOT NOTES BONUS			
Key: 28800128250,	amount: 111 oz,	Name: Dawn Glo Brand Tomato Ketchup			
Key: 28800128290,	amount: 113 oz,	Name: Farmer Girl Brand California Tomato Ketchup			
Key: 28800129523,	amount: 15.5 oz,	Name: Hanover Organic Pinto Beans			
Key: 28800129530,	amount: 15.5 oz,	Name: Hanover Organic Black Beans			
Key: 28800129547,	amount: 15.5 oz,	Name: Hanover Organic Light Red Kidney Beans			
Key: 28800129561,	amount: 15.5 oz,	Name: Hanover Organic Great Northern Beans			
Key: 28800132134,	amount: 41 OZ,	Name: HANOVER PORK N BEANS			
Key: 28800132462,	amount: 16 oz,	Name: Gibbs Pork & Beans In Tomato Sauce			
Key: 28800132501,	amount: 16 oz,	Name: Phillips Pork & Beans In Tomato Sauce			
Key: 28800132721,	amount: 7 lb (112oz),	Name: Dawn Glo Pork & Beans			
Key: 28800134305,	amount: 14.5 OZ,	Name: Hanover Pork & Beans In Tomato Sauce			
Key: 28800134336,	amount: 17.5 OZ,	Name: HAN PORK & BEANS BONUS			
Key: 28800134367,	amount: 110 OZ,	Name: WC HANOVER PORK & BEANS			
Key: 28800141440,	amount: 19.25 OZ,	Name: HAN BSB BAKED BEANS BONUS			
Key: 28800141457,	amount: 16 oz,	Name: Hanover Baked Beans Brown Sugar & Bacon			
Key: 28800141471,	Name: HANOVER BAKED BN				
Key: 28800141662,	amount: 40.50 OZ,	Name: HANOVER PINTO BEANS			
Key: 28800141686,	amount: 40.50 OZ,	Name: HANOVER BLK BEANS FAM			
Key: 28800141693,	amount: 40.50 OZ,	Name: HANOVER BLKEYE PE			

Figure 9: Insert() UPC



```

Search Results:
Key: 79, Name: INDIANA LOTTO
Key: 93, Name: treo 700w
Key: 123, Name: Wrsi Riversound cafe cd
Key: 161, Name: Dillons/Kroger Employee Coupon ($1.25 credit)
The key: 2140000070 does not exist in the BST
The key: 2140118461 does not exist in the BST
The key: 2144209103 does not exist in the BST
The key: 2144622711 does not exist in the BST
The key: 2147483647 does not exist in the BST
The key: 2158242769 does not exist in the BST
The key: 2158561631 does not exist in the BST
The key: 2158769549 does not exist in the BST
The key: 2160500567 does not exist in the BST
The key: 2172307284 does not exist in the BST
The key: 2177000074 does not exist in the BST
The key: 2184000098 does not exist in the BST
The key: 2187682888 does not exist in the BST
Search Tree Time: 7550800 nanoseconds, 7.5508 milliseconds, or 0.0075508 seconds
PS C:\Users\logan\OneDrive - UAB - The University of Alabama at Birmingham\UAB Files\FA 2023\CS-303\HW\CS-303-HW4>

```

Figure 10: searchTree() UPC 10

```

Search Results:
Key: 79, Name: INDIANA LOTTO
Key: 93, Name: treo 700w
Key: 123, Name: Wrsi Riversound cafe cd
Key: 161, Name: Dillons/Kroger Employee Coupon ($1.25 credit)
The key: 2140000070 does not exist in the BST
The key: 2140118461 does not exist in the BST
The key: 2144209103 does not exist in the BST
The key: 2144622711 does not exist in the BST
The key: 2147483647 does not exist in the BST
The key: 2158242769 does not exist in the BST
The key: 2158561631 does not exist in the BST
The key: 2158769549 does not exist in the BST
The key: 2160500567 does not exist in the BST
The key: 2172307284 does not exist in the BST
The key: 2177000074 does not exist in the BST
The key: 2184000098 does not exist in the BST
The key: 2187682888 does not exist in the BST
Search Tree Time: 4204200 nanoseconds, 4.2042 milliseconds, or 0.0042042 seconds
PS C:\Users\logan\OneDrive - UAB - The University of Alabama at Birmingham\UAB Files\FA 2023\CS-303\HW\CS-303-HW4>

```

Figure 11: searchTree() UPC 100

```

Search Results:
Key: 79, Name: INDIANA LOTTO
Key: 93, Name: treo 700w
Key: 123, Name: Wrsi Riversound cafe cd
Key: 161, Name: Dillons/Kroger Employee Coupon ($1.25 credit)
The key: 2140000070 does not exist in the BST
The key: 2140118461 does not exist in the BST
The key: 2144209103 does not exist in the BST
The key: 2144622711 does not exist in the BST
The key: 2147483647 does not exist in the BST
The key: 2158242769 does not exist in the BST
The key: 2158561631 does not exist in the BST
The key: 2158769549 does not exist in the BST
The key: 2160500567 does not exist in the BST
The key: 2172307284 does not exist in the BST
The key: 2177000074 does not exist in the BST
The key: 2184000098 does not exist in the BST
The key: 2187682888 does not exist in the BST
Search Tree Time: 3876400 nanoseconds, 3.8764 milliseconds, or 0.0038764 seconds
PS C:\Users\logan\OneDrive - UAB - The University of Alabama at Birmingham\UAB Files\FA 2023\CS-303\HW\CS-303-HW4>

```

Figure 12: searchTree() UPC 1K

```

Search Results:
Key: 79, Name: INDIANA LOTTO
Key: 93, Name: treo 700w
Key: 123, Name: Wrsi Riversound cafe cd
Key: 161, Name: Dillons/Kroger Employee Coupon ($1.25 credit)
The key: 2140000070 does not exist in the BST
The key: 2140118461 does not exist in the BST
The key: 2144209103 does not exist in the BST
The key: 2144622711 does not exist in the BST
The key: 2147483647 does not exist in the BST
The key: 2158242769 does not exist in the BST
The key: 2158561631 does not exist in the BST
The key: 2158769549 does not exist in the BST
The key: 2160500567 does not exist in the BST
The key: 2172307284 does not exist in the BST
The key: 2177000074 does not exist in the BST
The key: 2184000098 does not exist in the BST
The key: 2187682888 does not exist in the BST
Search Tree Time: 6478300 nanoseconds, 6.4783 milliseconds, or 0.0064783 seconds
PS C:\Users\logan\OneDrive - UAB - The University of Alabama at Birmingham\UAB Files\FA 2023\CS-303\HW\CS-303-HW4>

```

Figure 13: searchTree() UPC 10K

```

Search Results:
Key: 79, Name: INDIANA LOTTO
Key: 93, Name: treo 700w
Key: 123, Name: Wrsi Riversound cafe cd
Key: 161, Name: Dillons/Kroger Employee Coupon ($1.25 credit)
Key: 2140000070, Name: Rhinestone Watch
Key: 2140118461, Name: """"V""": Breakout/The Deception VHS Tape"
Key: 2144209103, amount: VHS, Name: Tintorera - Tiger Shark
Key: 2144622711, Name: Taxi : The Collector's Edition VHS
Key: 2147483647, Name: Toshiba 2805 DVD player
Key: 2158242769, amount: 288/1.12Z, Name: GREEN SUGAR COOKIES4276
Key: 2158561631, Name: HOT COCOA W/BKMK
Key: 2158769549, amount: njhjhn, Name: gjfhjbgkj
Key: 2160500567, amount: 2.25 oz (64)g, Name: Dollar Bar Rich Raspberry
Key: 2172307284, Name: Mixed seasonal flower bouquet
Key: 2177000074, Name: 4 way 13 AMP Extension Lead (Wilkinson UK)
Key: 2184000098, amount: 21 oz, Name: Christopher's Assorted Fruit Jellies
Key: 2187682888, Name: fairway
Search Tree Time: 4739400 nanoseconds, 4.7394 milliseconds, or 0.0047394 seconds
PS C:\Users\logan\OneDrive - UAB - The University of Alabama at Birmingham\UAB Files\FA 2023\CS-303\HW\CS-303-HW4>

```

Figure 14: searchTree() UPC 100K

```

Search Results:
Key: 79, Name: INDIANA LOTTO
Key: 93, Name: treo 700w
Key: 123, Name: Wrsi Riversound cafe cd
Key: 161, Name: Dillons/Kroger Employee Coupon ($1.25 credit)
Key: 2140000070, Name: Rhinestone Watch
Key: 2140118461, Name: """"V""": Breakout/The Deception VHS Tape"
Key: 2144209103, amount: VHS, Name: Tintorera - Tiger Shark
Key: 2144622711, Name: Taxi : The Collector's Edition VHS
Key: 2147483647, Name: Toshiba 2805 DVD player
Key: 2158242769, amount: 288/1.12Z, Name: GREEN SUGAR COOKIES4276
Key: 2158561631, Name: HOT COCOA W/BKMK
Key: 2158769549, amount: njhjhn, Name: gjfhjbgkj
Key: 2160500567, amount: 2.25 oz (64)g, Name: Dollar Bar Rich Raspberry
Key: 2172307284, Name: Mixed seasonal flower bouquet
Key: 2177000074, Name: 4 way 13 AMP Extension Lead (Wilkinson UK)
Key: 2184000098, amount: 21 oz, Name: Christopher's Assorted Fruit Jellies
Key: 2187682888, Name: fairway
Search Tree Time: 4361600 nanoseconds, 4.3616 milliseconds, or 0.0043616 seconds
PS C:\Users\logan\OneDrive - UAB - The University of Alabama at Birmingham\UAB Files\FA 2023\CS-303\HW\CS-303-HW4> 

```

Figure 15: searchTree() UPC



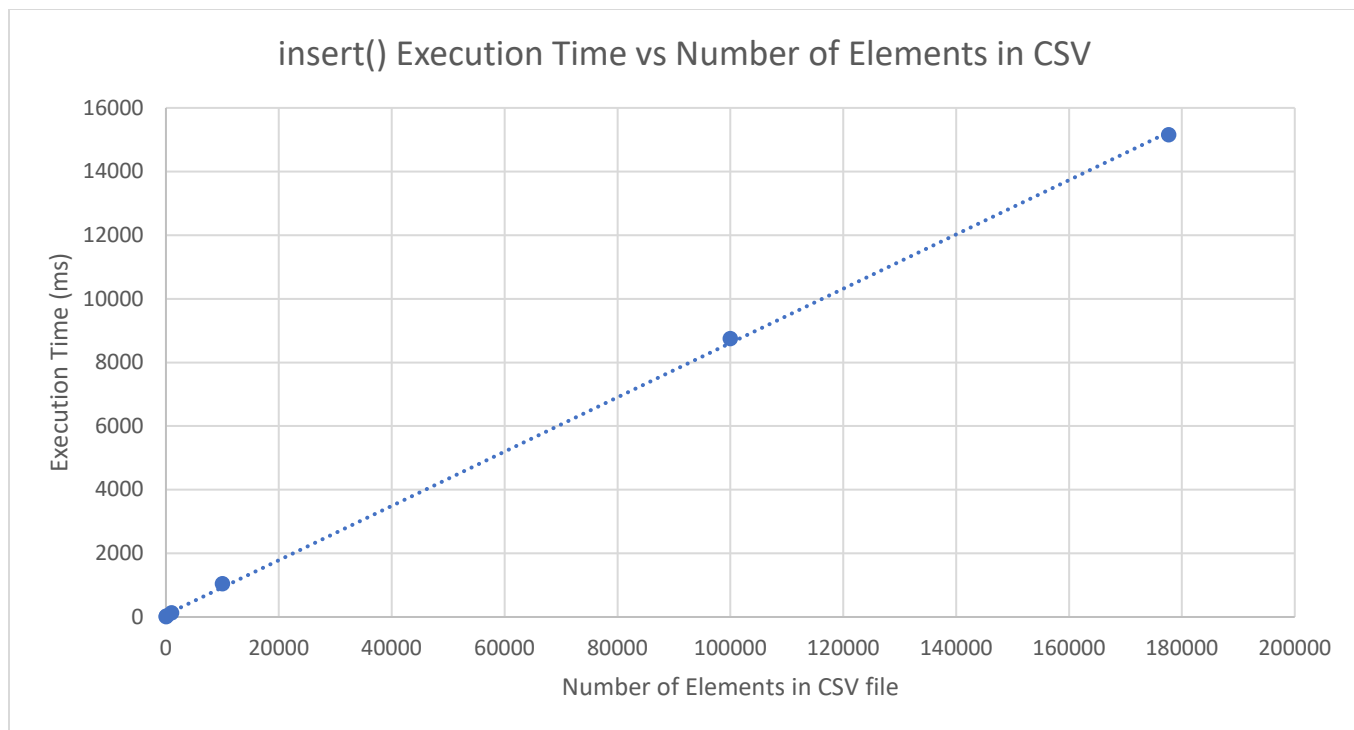


Figure 16: insert() Execution Time vs Number of Elements in CSV

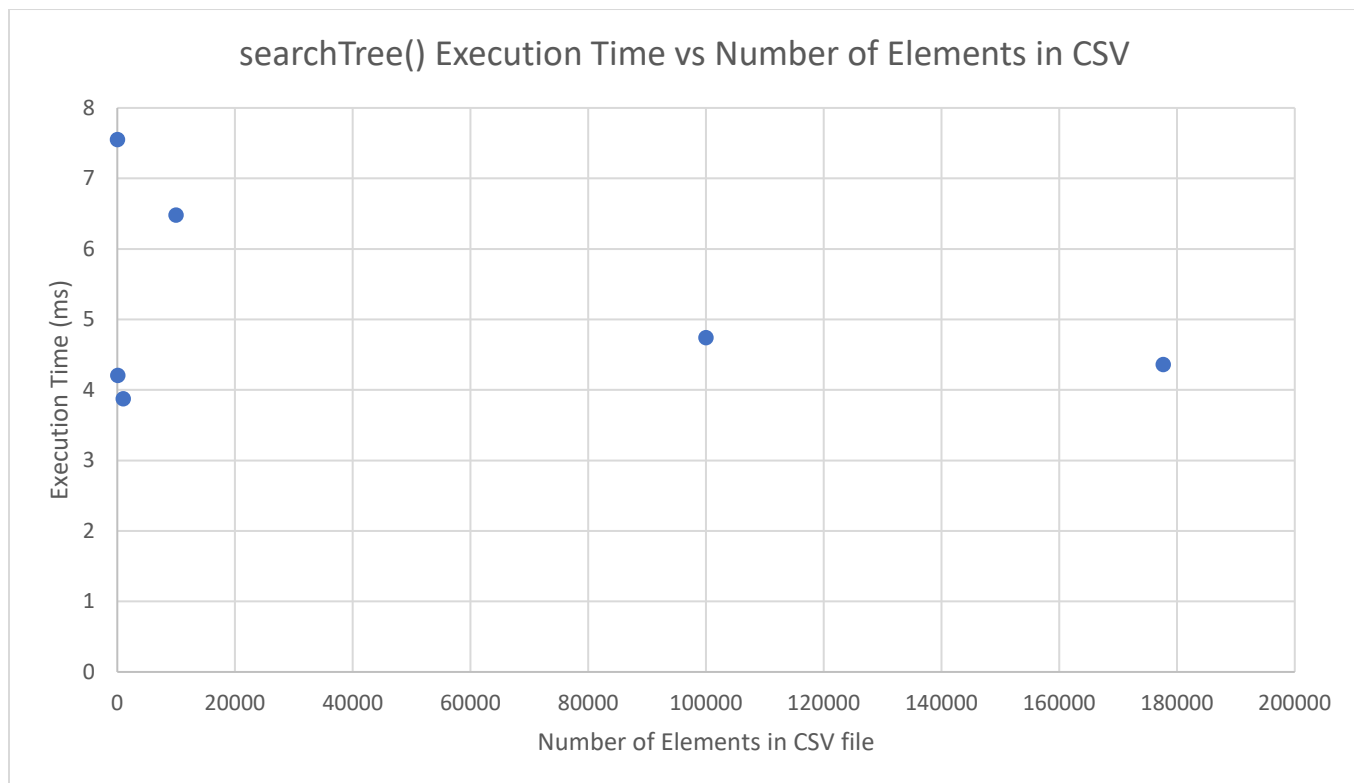


Figure 17: searchTree() Execution Time vs Number of Elements in CSV

The data above clearly shows that the time complexity of the insert() function is linear, or  $O(n)$ . The time complexity of the search() function is not so easily discernable, however, as just by looking at the graph it would seem there is no connection between the execution time of the algorithm and the number of elements operated on. This isn't reflective of the algorithm's actual time complexity, which is  $O(\log(n))$ . This is because the algorithm is similar to binary search, in that it divides the tree that needs to be searched in half with each iteration, resulting in a time complexity of  $O(\log(n))$  on average and  $O(n)$  in the worst case. The reason that the graph poorly reflects the actual time complexity of the algorithm could be due to the scaling of the datasets or the fact that the same search keys were used for each test. In the future, it could be more effective to randomize the keys with each test.

## 4. Sources

<https://stackoverflow.com/questions/64488594/reading-from-csv-file-and-create-object>

<https://www.youtube.com/watch?v=2gZYhlhyDyA>

<https://www.geeksforgeeks.org/binary-search-tree-traversal-inorder-preorder-post-order/>

<https://www.geeksforgeeks.org/properties-of-binary-tree/>

<https://www.geeksforgeeks.org/inorder-tree-traversal-without-recursion/>

<https://chat.openai.com/c/78371716-dd16-491e-a6b8-e9bc5faa34d9>