In this lab, I will be working with binary search trees and different ways that they could be used for instance, to find the value in the tree, convert the values of a tree to a list, build a sorted tree, and to find the elements at a depth in a tree.

**Question 1**

**Objective**

For question 1, my objective is to be able to display the binary search tree by creating a figure of a tree and inputting the values of the tree in their respective positions. In the figure I will also need to create circles that will hold the values of the tree after the end of each line in the figure.

**Proposed solution**

For this problem I tried to combine the drawTrees method and draw\_Circles method from lab 1 since I knew I needed those figures to create a tree. However, when trying to combine these two methods, I would have issues on trying to place a circle inside my tree and I found out that it was going to be a larger issue to deal with since they had various different variables to implement. Also, another issue was that I wasn’t able to implement the values of my tree into the figure correctly since they would be placed with the circles that wouldn’t appear on my tree.

The second solution that I came up with was to re-work my drawTrees method in order to be able to create a new was to implement the circles inside the figure. The way I planned to re-work my drawTrees method was to make a second method that draws the shape of a triangle since a tree consists of various triangles. Also, in my drawTrees method I needed to implement a way to insert the circles in the figure while containing the values that came from the tree.

**Implementation**

The way I began to re-work my code was to refer to the drawSquares method from lab 1 and modify the list i1 into making it have only two points which are 1 and 0. The purpose of using this is to be able to create the shape of a tree which consists of triangles and in this instance, I don’t close the triangles since I need them to keep drawing tree over and over; Once Modifying the lab 1 drawSquares method is done, I changed its name to draw\_Triangle to fit the its function. Additionally, I didn’t change anything else with that method since it functioned properly. After that I had to look at my drawTrees method and modify the original code, to begin modifying I changed the parameters of my method and passed the Tree, my x and y coordinates, and xMove and yMove which are the directions that my x and y will be moving.

Inside the drawTrees method I now had to traverse my tree until its empty and since we always start at the root I declared my line of code if Tree is not None. Once that if is made I implemented a line of code that allows me to implement a circle, but to do that I needed to use my x value and subtract it by .4 since the circle would look uneven in the figure, I had to add the y coordinate to the yMove since we need to place the circle at the top, after that I was able to implement the current item of the tree into my circle. After implementing the value, I needed to input the values for the left side and right side of my tree.

When I began to implement the values of the left side, I implemented an if statement that will traverse the left side of my tree until it reaches None. Inside the if statement I used some of the original code from the drawTrees method, I used the variable q and declared it to my plot which is np.array and inside it would subtract the x from its xMove since in the tree we have to move left and draw that way, however in that array I made another plot point where it draws in the y axis by adding y to the yMove variable which will begin to draw at that point. After making the plot to draw, I needed to call the draw\_Triangle method in order to begin drawing the shapes for my figure, but when calling the method, I passed as my parameters ax, 1 which is the amount of times I need to draw the shape, q which contains the plot points, and .9 as the weight of the drawing. After calling that method, I needed to use recursion to call my drawTrees method again in order to keep traversing through the tree and inserting the values in the figure. When calling recursively, I passed the left side of the tree, subtracted x from xMove and y from yMove in order to be able to move left and down, then I divided xMove by 2 , and passed yMove without affecting it. Assuming this works the method will keep going through the tree and insert all of the left side variables.

Now I need to implement the right side of my tree into the figure and its actually similar to the way the left side was implemented. I first have to make an if statement that traverses the right side until it reaches none. Inside the if I declared variable q1 to the np.array and instead of subtracting the x first, I actually added the y to the yMove first and in the other list I added the x and xMove since we need to draw to the right side of the figure which creates the plot. After that we need to call the draw\_Triangles method like before in order to create the shapes for our drawing. Then I need to call the method recursively and pass the right side of my tree, and the x and xMove since we need to move right, subtract y from yMove since we move down in a tree, we still divide 2 from xMove, and pass the yMove variable without any changes to it.

**Tracing**

To trace this my tree will be using the sorted list with the variables [10,4,2,8,15,12,18] and when calling drawTrees I passed the tree and the values for x,y,xMove,yMove as 10. When the method is called it begins to check if the root of the tree is not none and since it had 10 as the root then it passes the base case. Inside the base case the program inserts the value 10 into a circle into the top of the tree. Now that the root is drawn, the code begins to check if the left side of my tree is not none and since my left has the variable 4 then it goes into the if statement. In the statement the variables x and xMove are subtracted making it 0 and leaves y unaffected, but in the second part y is added with yMove which gives us 0 and the x is unaffected. After these values are made to begin plotting the method draw\_Triangle is called and draws the left side of the triangle. After the line is drawn then the method is called recursively an moves on to the next node of the left side of the tree, subtracts the x and xMove making x 0, then y is subtracted from yMove making y 0 , then xMove is divided by 2 which makes xMove 5 and yMove is unaffected. Since its calls the method by passing the left side of the tree and the new value that is printed is 4 since its reading it as the root, but for that position it is in only. Now that its printed it will go into the if statement for the left side since it’s still not empty and process the work that was done before and draw another left triangle in the tree. Once the drawing is done then the code recursively calls the method again and prints the value 2, however since this is the last value of the left side then it stops going through the code for that value. Referring to the value 4 we can see that it still has another number that is 8 which is on the right side, meaning that this time the code will run the if statement for if the right side of the tree is not none and since it has the value 8 then it begins to draw the right side of the triangle as the left side would do. However, when calling the method recursively the x value is added to the xMove since its moving to the right. When the method is called again then the code prints the value 8 only for that position but doesn’t do anything else since it doesn’t fit the if statements. Since there are variables still missing the original root of the tree had variables on the right which are 15,12, and 18. Since the right side is not empty then the if statement draws the right side of the triangle and calls the method recursively and prints out 15, also this is done for the value 18 later on in the code. Since 15 has a left side then the code draws a left triangle and calls the method in order to print out 12 and since 12 is the last value then it stops going the same applies for the value 18.

A close up of a mans face

Description automatically generated

**Question 2**

**Objective**

For this question I’m supposed to make an iterative method that searches for a value in the binary search tree. By iterative means that it needs to use loops and no recursion.

**Proposed solution**

For this method, I know that I need to use loops to traverse the tree and I need to return if the number that I was given was found in the tree. So, to make sure I find the value I need to compare the current item in the tree to the value. If the item in the tree is not found when comparing to the current item then I need to see if its less than or greater than the current item and traverse to the left of the tree if its less than or traverse through the right if its greater than to the value we are looking for, until the value is found then it prints that it found the value.

**Implementation**

for this method I will be passing two variables which is the tree and k which is a number that may or may not be in the list. In the method I want to declare temp to T just to make sure the original is not affected. In order to loop through the tree, I used a while loop and while its True then the loop iterates through the tree until its false, also looking at this the code can be changed to while temp is not None. In the method I made an if statement that if the item in my temporary tree is equals to k then in prints that it found the value and returns the tree. After that if I made an elif that compares the temp current item to the k and here its if the item in temp is less than k then temp is now declared as temp.right to traverse to the right of the tree. There’s another elif that if the item in my temp is greater than k then the temp is now declared as temp.left to traverse the left of the tree.

**Tracing**

For this trace I will be looking at the tree that has the values [10,4,2,8,15,12,18] and I will be searching for the value 4. When the tree and 4 is passed to the search method, the code stores the tree in a temporary tree do avoid changes to it. The code then moves on to the while loop and since its true that its not empty then it compares the value at the root of the tree to k which is 4 and if they’re equal then it prints that it found the value and returns the temp. however since our root is not equal to 4 because the root is 10, the code moves on to the next if and if the item that is currently the root is less than the k then the temp moves to the right side of the tree. Since 10> 4, the only option left is the third if because if the item at the temp is less than k then the temp moves to the left value. Now in the while loop the number that is looked as the root is