

Homework 2

1. For the last two weeks, we discussed:
 - a. Binary Search Trees
 - i. Creation
 - ii. Search - $O(n)$ worst case, $O(\log n)$ or $O(\text{Height})$ usually
 - iii. Insertion (smaller numbers to the left, greater to the right)
 - iv. Deletions
 1. Leaf
 2. Parent node (replace with right, all the way to the left)
 - b. Problems with Binary trees and introduction to Balanced Search Trees
 - c. Converting to a Balanced tree using an intermediate sorted list
 - d. Introduction to 2-3 trees
 - i. 2 node, 3 node
 - ii. Insertions
 1. 2 node
 2. 3 node
 3. 3 node with a 3 node parent

If you missed either week, you can review the material here (ignore the java code):

<https://algs4.cs.princeton.edu/32bst/> for Binary Search Trees and

<https://algs4.cs.princeton.edu/33balanced/> for Balanced Search Trees (up to and including the section called "*Splitting the root.*")

For homework,

1. Try and implement the:
 - a. creation and insertion into a binary tree. You will need to implement a class Node that holds a value, and a left and right pointer that also points to null or another Node.
 - b. Try to implement a search function in the class as well

Try and accomplish this for the weekend after Thanksgiving Break

2. AFTER reading the section on 2-3 trees, on paper:
 - a. Create a balanced binary tree with the letters in "school nova at stony brook" (eliminate duplicates). Hint: create a sorted list first
 - b. Convert this to a perfectly balanced 2-3 tree
 - c. Insert the values d, u and y, maintaining a balanced 2-3 tree

Node:

```
data
left
right
```

```
insert(root, value):
```

```
    if root is null:
        create new node with value
        return new node
    else if value < root.data:
        root.left = insert(root.left, value)
    else:
        root.right = insert(root.right, value)
    return root
```

```
search(root, value):
```

```
    if root is null or root.data == value:
        return root
    else if value < root.data:
        return search(root.left, value)
    else:
        return search(root.right, value)
```

HW Policy:

As mentioned in class there are plenty of sites on the internet (including generative AI tools) with descriptions of the algorithms, and in many cases, the solutions to this problem. To maximize learning, feel free to use resources to review the material discussed in class, but attempt to write the code on your own. The exercises will strengthen your understanding of the material, and Python in general.