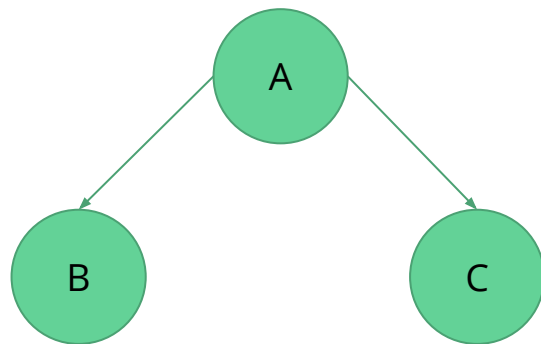


Tree and Heap Sort

Leap@CMU 2017

Sorting Binary Trees

1. Using your previous project, create a binary tree from unordered data
2. Then traverse through the tree in such a way that you sort the data



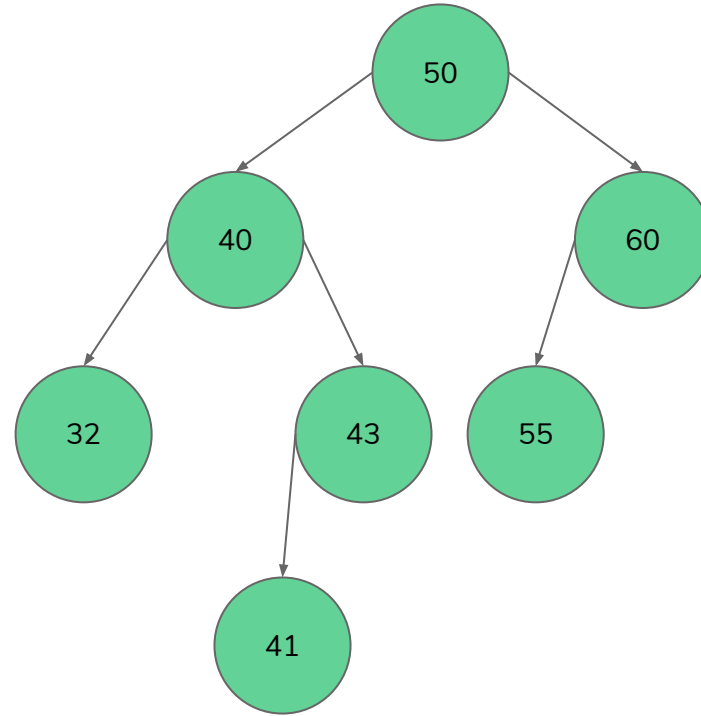
This sorts to: [B, A, C]

We can use recursion to traverse the left subtree, then the root, then the right subtree. This will sort the data.

Worked Example

This sorts into:

[32, 40, 41, 43, 50, 55, 60]



Creating Heaps

The simplest way to create heaps is to transform your binary tree

1. Take your binary tree
2. Compare each node with its parent and children and swap them if need be (to achieve either a max-heap or min-heap)
3. Repeat until done

Sorting Heaps

Assume that your heap is structured with the largest element as the root, and you're sorting in descending order.

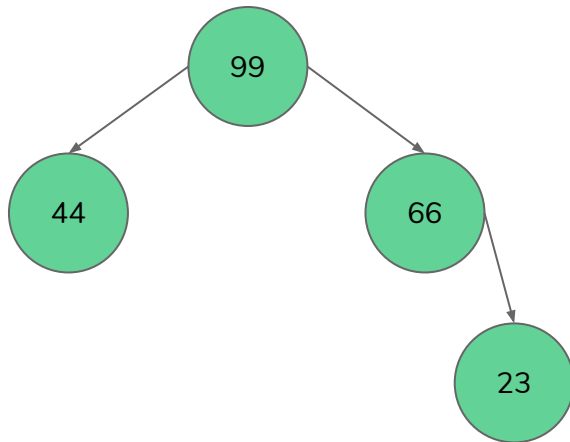
1. Extract the root and place into your sorted list
2. Replace the root as `null`
3. Compare the `null` node with its children, take the largest child and swap it with the `null` node. Repeat this until the `null` node is at the bottom of the tree.
4. Repeat steps 1 - 4 until all nodes are `null`.

Worked Example

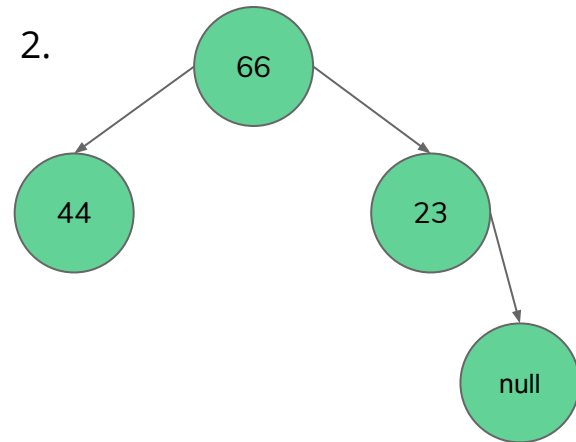
This heap sorts to:

1. [99]
2. [99, 66]
3. [99, 66, 44]
4. [99, 66, 44, 23]

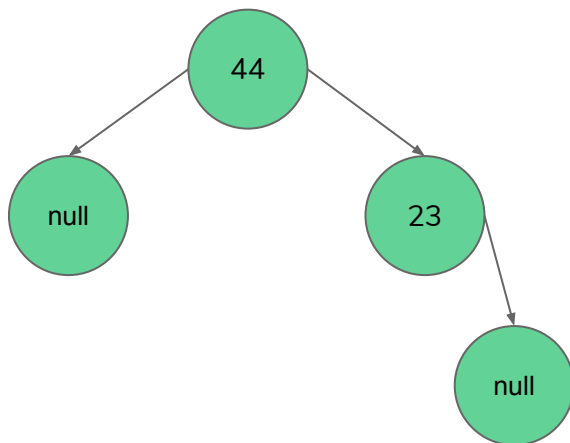
1.



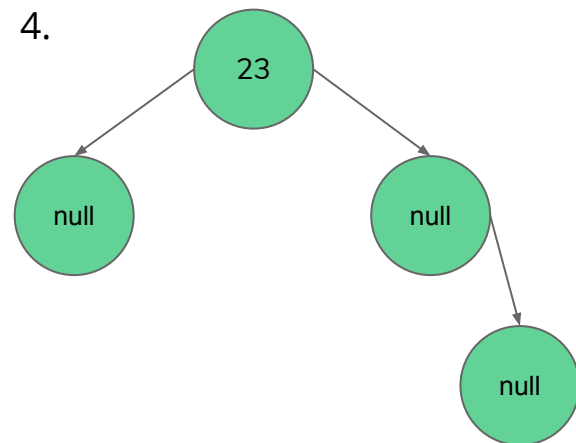
2.



3.



4.



Project Guidelines

[*] = Implement Binary Sort

[**] = Implement Heap Sort

Note: Your Unit 4.3 projects needs to be completed for these methods to be implemented properly.