name: Jason McKinnerney

CS3358 Final, Summer 2021

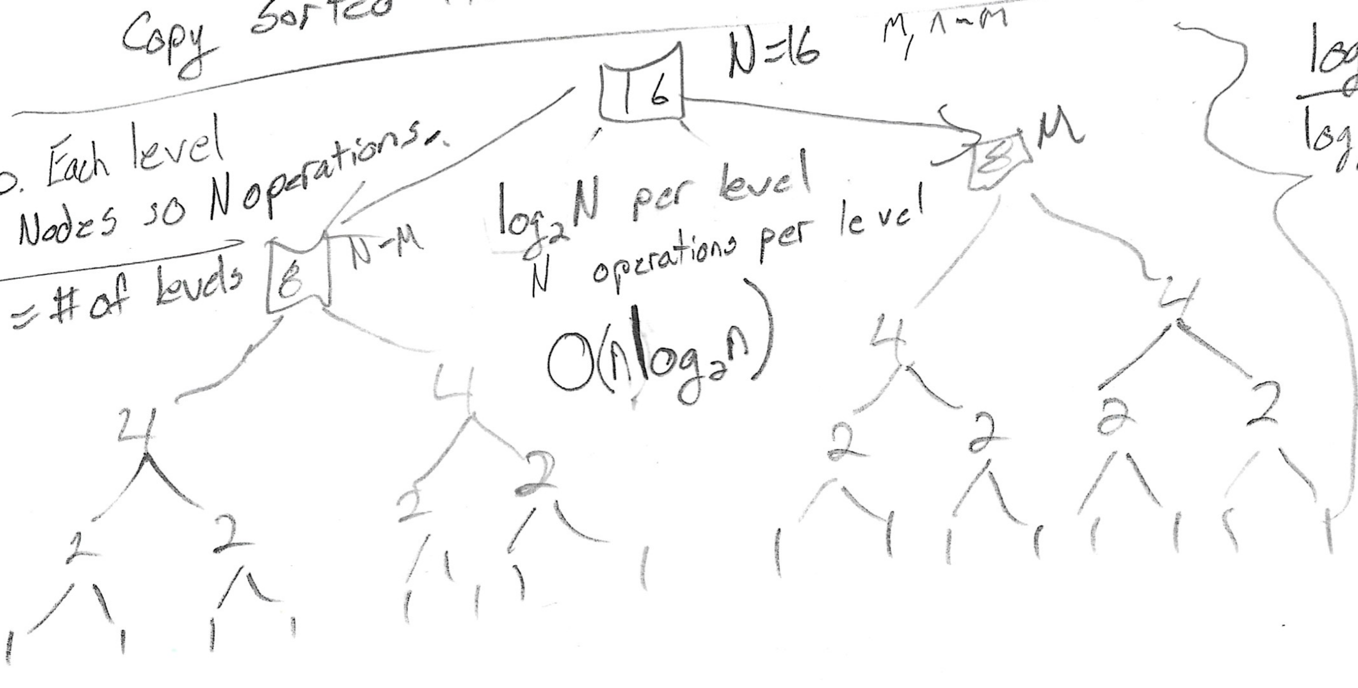
1. (25 points)
   a) (10 points) Provide pseudocode for recursive MergeSort function
   b) (15 points) Prove the upper asymptotic bound for MergeSort using the recursion tree as was done in class (and in the textbook).

a.

Set SaveFirst to LeftFirst
Set Index to LeftFirst

While{(More items in left half AND More items in Right half)
    if values[leftfirst] < values[rightFirst]
        Set tempArray[index] to values[leftfirst]
        increment leftfirst
    else
        Set tempArray[index] to values[rightFirst]
        increment right first

Increment Index
}
Copy remaining items from left half to tempArray
Copy remaining items from right half to tempArray
Copy Sorted items from temp Array back to Values



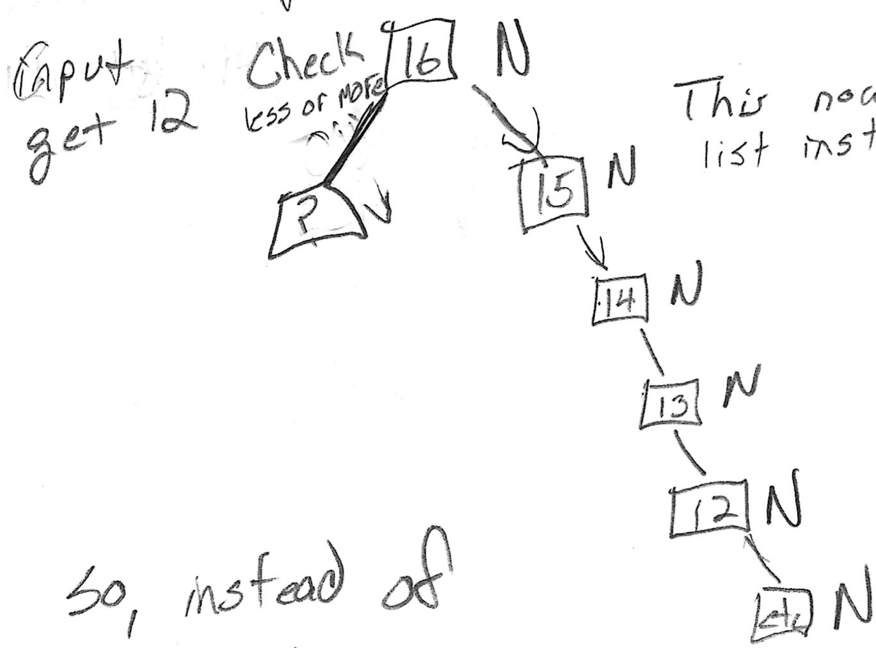
b. Each level do all Nodes so N operations.
$\log_2 N$ = # of levels
$\log_2 N$ per level
N operations per level
$O(n \log_2 n)$

$\log_2 N$
$\log_2 16$
$= 4$ levels of Merging

name: Jason McKinnerney

2. (25 points) Explain how and due to what kind of input a binary search tree (BST) can degenerate and explain what the upper asymptotic bound for GetItem function of BST becomes in this case (provide both the formula and explanation).

If an already sorted list is given as input into a binary search tree, then all nodes will go onto one side of the tree.



This now behaves as a linked list instead of a binary tree

So, instead of

$O(\log_2 N)$ by splitting the nodes in half each time,

A comparison is needed for each node, making $O(N)$

3. (25 points)

a) Perform the Selection sort algorithm writing out the intermediate states of the array as is done in the textbook starting from the input array below

b) Prove the upper asymptotic bound of the Selection sort using the summation as was done in class and in the textbook

	a values	b	c	d	e
[0]	126	1	1	1	1
[1]	43	43	26	26	26
[2]	26	26	43	43	43
[3]	1	126	126	126	113
[4]	113	113	113	113	126

✓ sorted

search for next then swap

$$(N-1)+(N-2)+(N-3)+(N-4)+(N-5)$$

$$1 + 2 + 3 + 4 + N-1$$

$$\frac{N + N + N + N + 1}{N-1}$$

$$\frac{N(N-1)}{2}$$

$$= \frac{n^2}{2} - \cancel{\frac{n}{2}}$$

$$\frac{n^2}{2} < n^2$$

$$\boxed{\text{so } O(n^2)}$$

name: Jason McKinnerney

4. (25 points)
   a) (10 points) Write the pseudocode of the inorder walk function of a binary search tree.
   b) (10 points) Derive the upper asymptotic bound for the inorder function
   c) (5 points) What is the inorder walk used for (i.e. how does it arrange the keys of the nodes) ?

The action on each node should be:
inQue.Enqueue(tree->info);

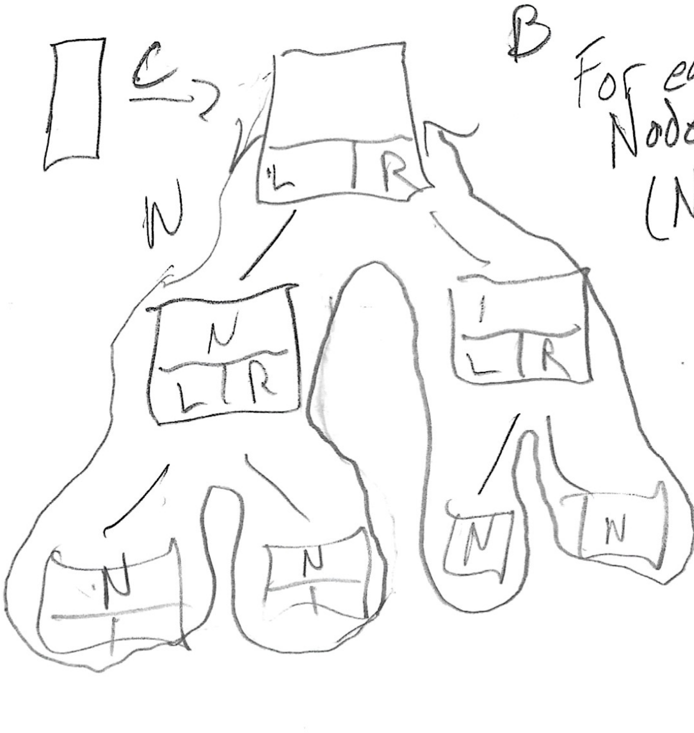
Remember about the base case (termination condition).

void InOrder(TreeNode* tree, QueType& inQue)
// Post: inQue contains the tree items in inorder.

a.
```
{
  if (tree != Null)
  {
    InOrder(tree->left, inQue);
    inQue.Enque(tree->info);
    InOrder(tree->right, inQue);
  }
}
```

c. An inorder traversal of a binary search tree arranges the key in ascending key order.
Increasing value of keys

B



For each Node: (N)

Check if null
Left Print
Right Print
Return

Possible 4 operations
4N = 4N
worst case

$4N \leq O(N)$

so

$O(N)$