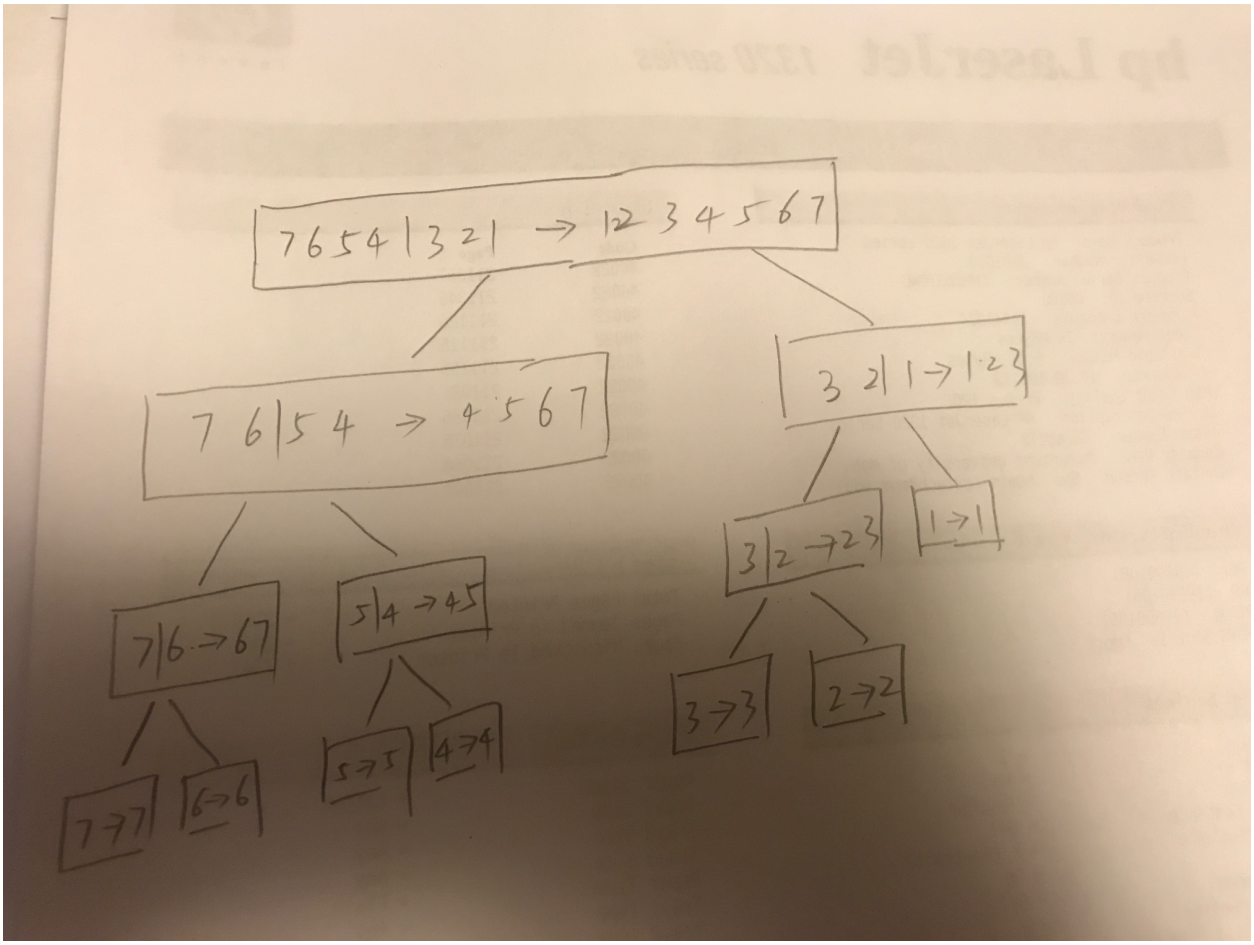
Yuliang Jin 986381

Problem 1

Determine whether InsertionSort, BubbleSort, SelectionSort are stable sorting algorithms, and in each case, explain your answer.

InsertionSort is stable, because the process if from front to the end so same values maintains the sequence after the sorting. BubbleSort is also stable because two equal elements will not swap. SelectionSort is also stable because the algorithm does not swap for two equal values when find the next small value.

Problem2

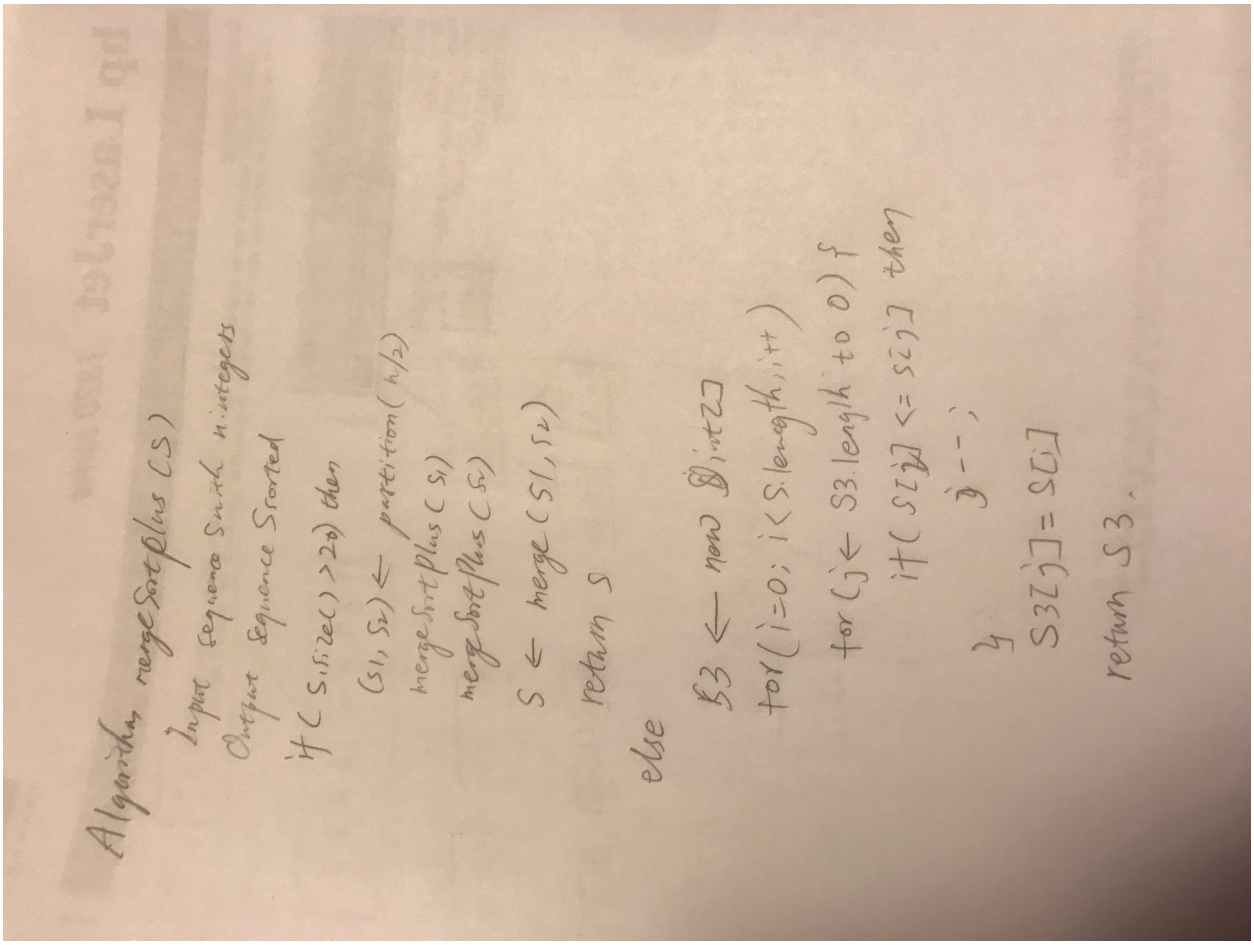


Problem 3

Sometimes MergeSort is supplemented with a secondary sorting routine (typically, InsertionSort is used) in the following way: During the recursion in MergeSort, the size of the array being sorted becomes smaller and smaller. To create a hybrid sorting routine, when a recursive call requires the algorithm to process an array with 20 or fewer elements, give this array to InsertionSort and patch in the result after it has finished. Call this hybrid algorithm MergeSortPlus.

pseudo code

Express the steps of MergeSortPlus in the pseudo-code language we are using in class



Code

```
void mergeSort(int[] tempStorage, int lower, int upper) {
                if(lower==upper){
                        return;
                }

                //Use InsertionSort to process
                if(upper < lower + 20) {
```

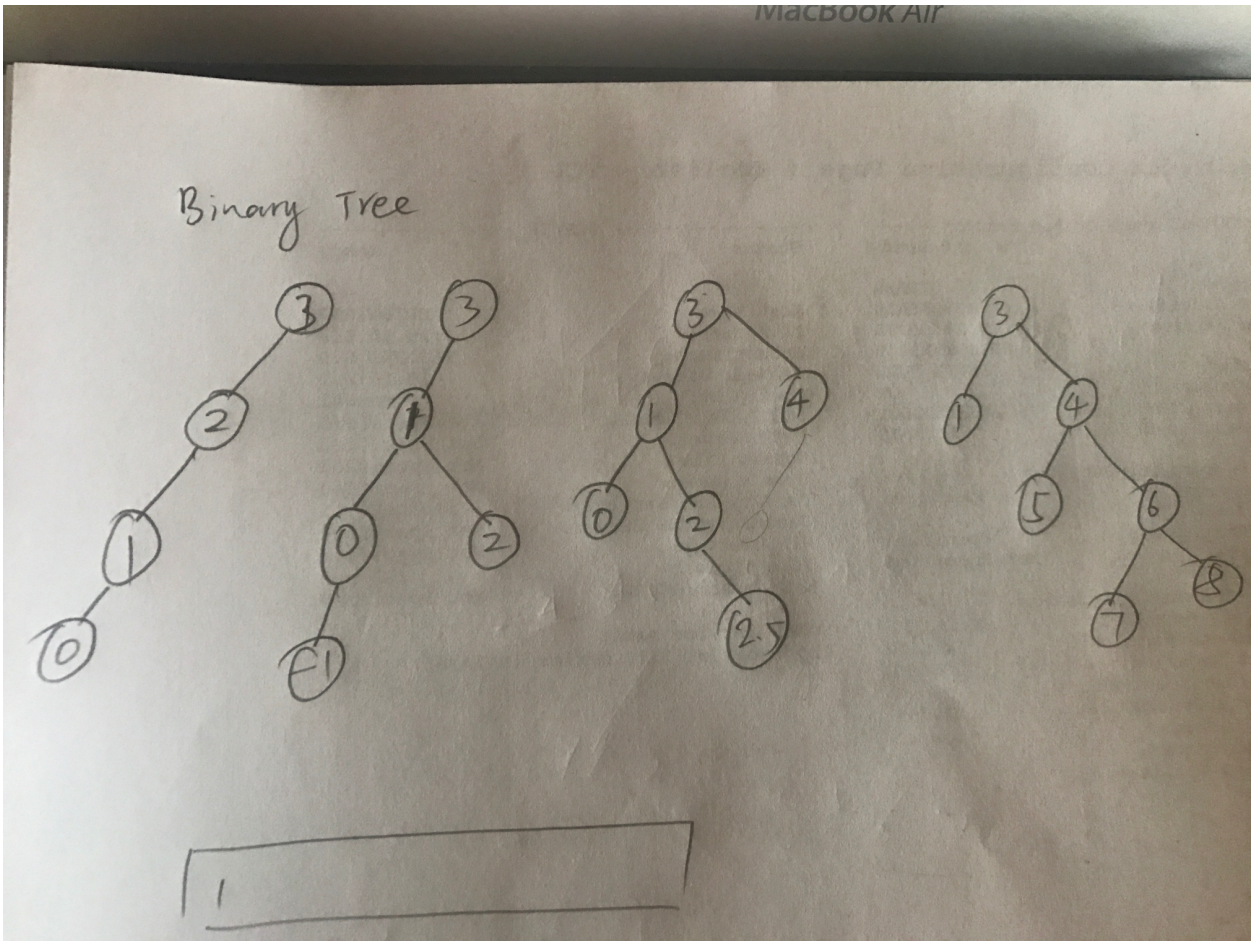
```
                        int len = upper - lower + 1;
                        int temp = 0;
                        int j = 0;
                        for(int i = 1; i < len; ++i) {
                                temp = tempStorage[i];
                                j=i;
                                while(j>0 && temp < tempStorage[j-1]){
                                        tempStorage[j] = tempStorage[j-1];
                                        j--;
                                }
                                tempStorage[j]=temp;
                        }
                        return;
                }

                else {
                        int mid = (lower+upper)/2;
                        mergeSort(tempStorage,lower,mid);  //sort left half
                        mergeSort(tempStorage,mid+1, upper); //sort right half
                        merge(tempStorage,lower,mid+1,upper); //merge them
                }
        }
```

Comparison

The MergeSortPlus is more efficient than the MergeSort because for array less than 20 elements the InsertionSort is better than MergeSort.

Binary Trees. A binary tree is a tree in which every node has at most two children.

Write out 4 different binary trees, each having height = 3 – make sure that no two of your trees have the same number of nodes. (There is no need to give labels to the nodes.)



Examine the trees you have drawn and decide whether the following statement is true or false:

Every binary tree of height 3 has at most 2^3=8 leaves. This is true. Because in a binary tree, each node has at most two child node. So a tree with height 3 has at most 2^3 = 8 nodes.

Based on your answer to b, what do you think is true in general about the number of leaves of a binary tree of height n?

2 ^ n