Mariya Eggensperger
CST 370, Spring 2017
Dr. Feiling Jia
Design/Analysis
of Algorithms

**CST 370 Homework (Sorting and Trees)**

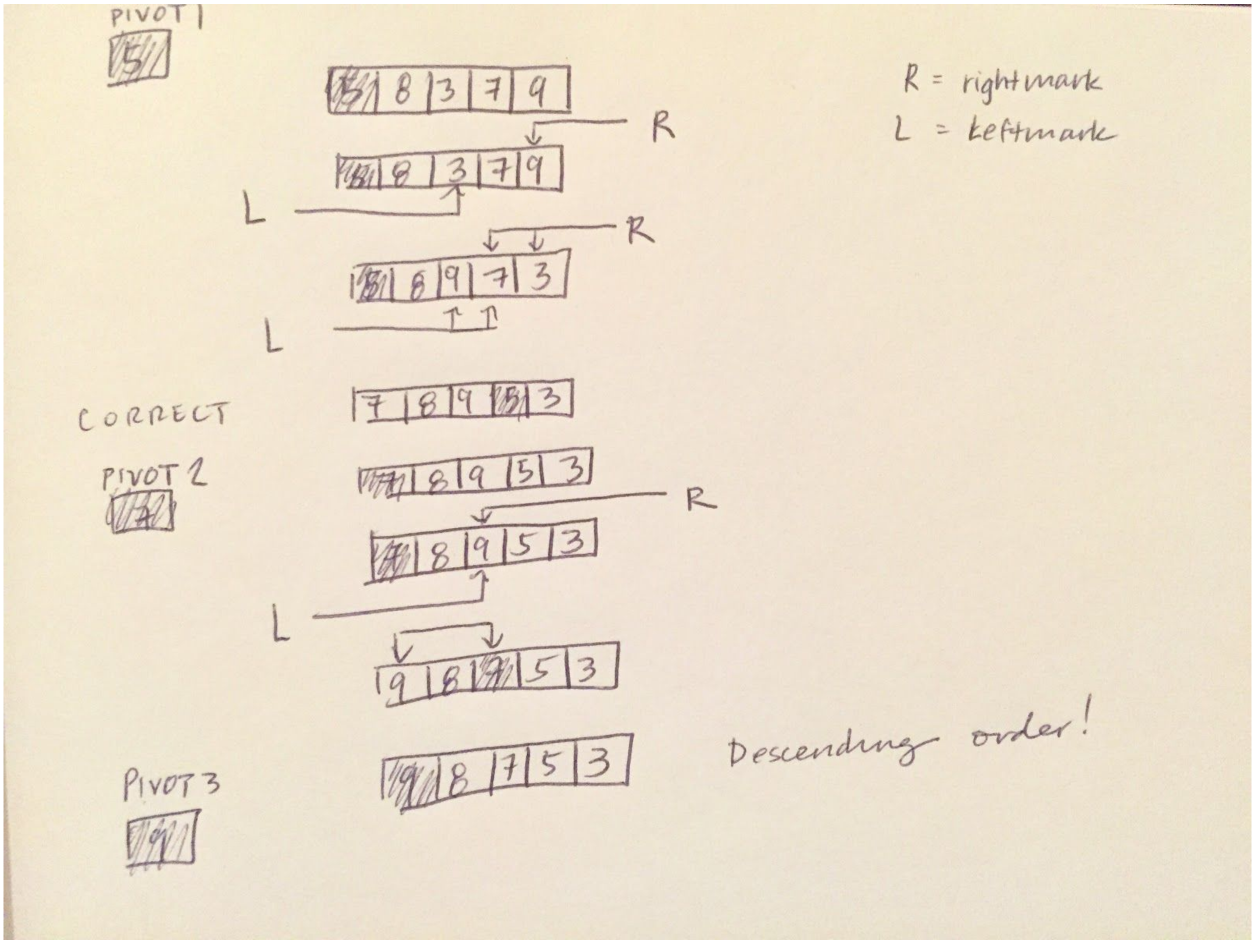
1. (25 points) Consider the unsorted array: 92, 81, 73, 44, 35, 56, 89, 68. Imagine that you run a sorting algorithm on the array, and at some moment in time, the array is ordered as: 81, 92, 73, 44, 35, 56, 68, 89. Which of the following sorting algorithms may have been running? Explain your answer.

- Quicksort (1st item as pivot)
- ***Mergesort***
- Insertion sort
- Selection sort
-

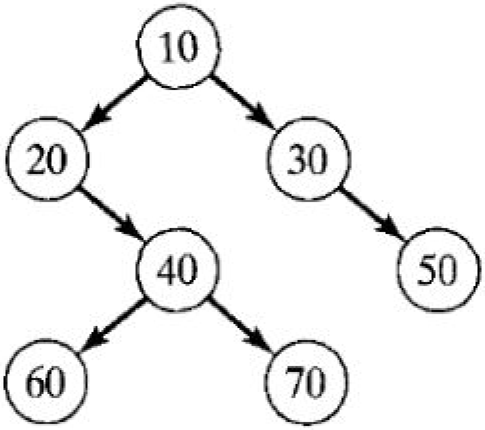
This is a merge sort; the merge sort is a recursive algorithm that splits a list in half. If the list is empty or has one element, by definition, it is sorted. If the list has more than one item, it is split and recursively invoke a merge sort on both halves. Once both halves have been sorted, the final operation merge is performed. In merging, two smaller sorted lists are combined into a single, sorted, new list.

2. (25 points) Consider the following unsorted arrays of numbers: 5, 8, 3, 7, 9. Obtain the numbers in sorted order (descending) by applying the quick sort algorithm. You are required to outline each step of the algorithm (i.e., show the state of the list after each iteration of the algorithm). Assume that the first element is always picked as the “pivot”.

**Please note that the image below should have 3 pivots: 5, 7, 9. See image for answer.



3. (25 points) Is the following tree a binary tree? If so, is it a binary search tree? Explain your answer.



**Binary Tree : YES**
A binary tree is made of nodes, where each node contains a left pointer, a right pointer, and a data element. The root pointer points to the topmost node in the tree, as is the case in this image. The left and right pointers recursively point to smaller subtrees on either side.

**Binary Search Tree: NO**
However, this is not a binary search tree due to failing elements. A *binary search tree* is an ordered *binary tree,* or a type of binary tree where the nodes are arranged in order. All of the elements in its left subtree are less-or-equal to the node (<=); whereas, all the elements in the right subtree are greater than the node ( >). In both cases, this is not true respectively with the image.

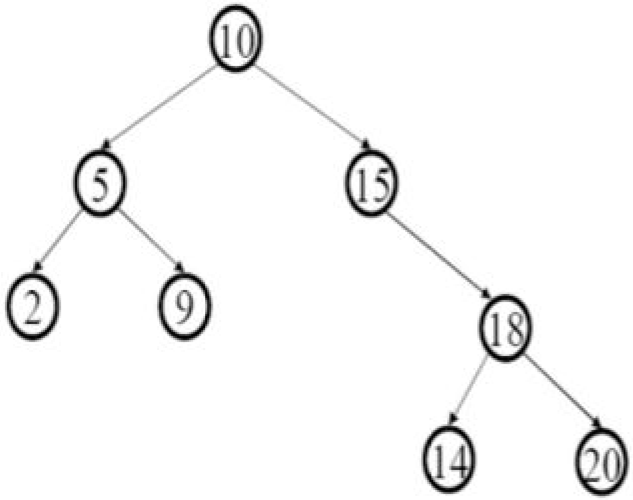
4. (25 points) Is the following tree a binary search tree? Explain your answer.

NO, this is a binary search tree:

- The left subtree of a node contains only nodes with keys less than the node's key.
- The right subtree of a node contains only nodes with keys greater than the node's key.
- The left and right subtree each must also be a binary search tree.

***** HERE Is where the flaw occurs; this image is not a binary search tree because the right side of the image has a binary tree which does not fulfill the binary tree requirement.***

- There must be no duplicate nodes.


10
5
15
2
9
18
14
20