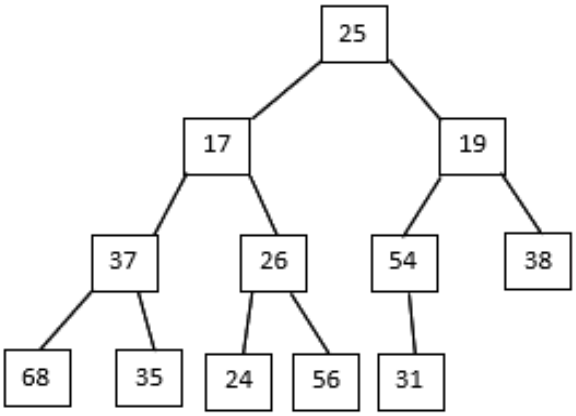
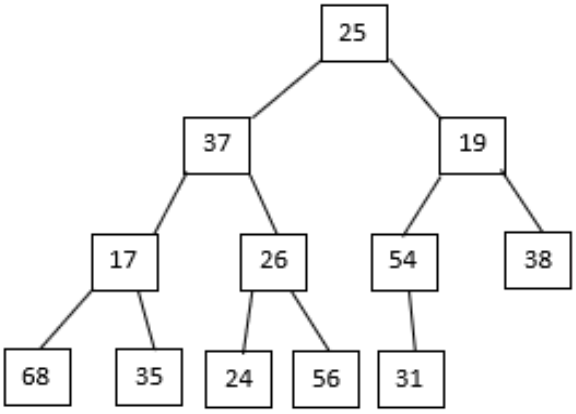
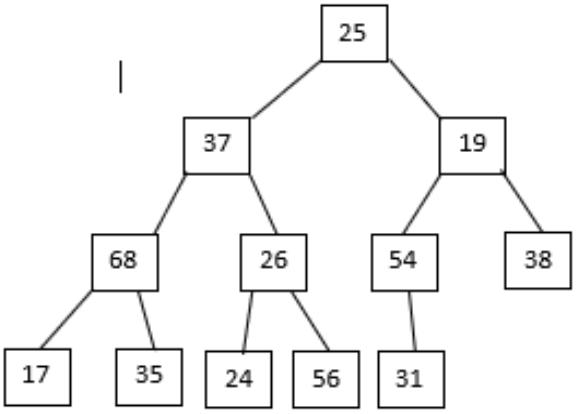
Vincent Latona
September 21, 2021

**Assignment 4**

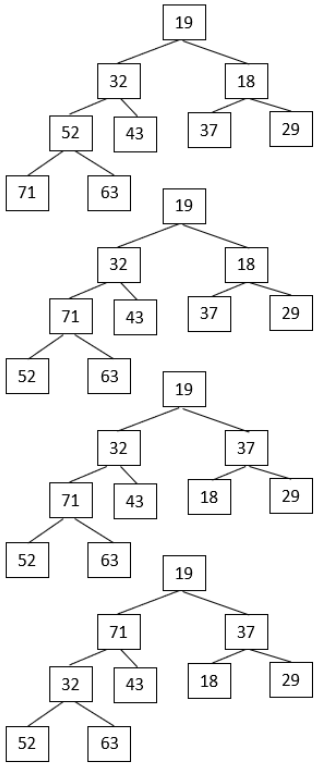
1. Below are the trees for each element swap in problem 1

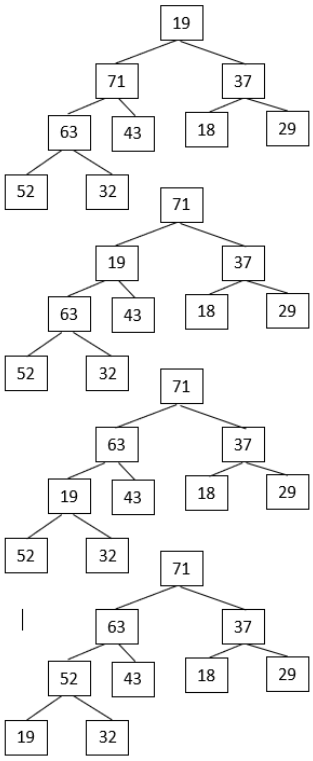


2. Below are the trees for each element swap in problem 2



3. **Loop Invariant** - At the start of each iteration, the left sub-array A contains the elements A[1..i-1] originally contained in the sub-array and a count is kept of the number of elements that are greater-than-or-equal-to the first element A[1] for the entire array.

   **Initialization** - Before the first iteration, the sub-array contains the first element A[1], which is equal to the first element, thus the count is initialized to 1. Thus, the loop invariant is true.

   **Maintenance** - Assume that the sub-array still contains the values A[1..i-1] and a running-count has been kept. In the loop, we iterate to the next element and evaluate whether A[i] is greater-than-or-equal-to A[1]. If the evaluation is true, then the running-count is incremented; otherwise, the loop will either terminate or reiterate. The sub-array contains the elements from A[1..i-1] and a running-count of elements greater-than-or-equal-to A[1] is maintained, thus the loop invariant is true after an iteration.

   **Termination** - The loop terminates when $i > n$, thus $i = n+1$. This means that the sub-array contains all elements of A[1..n] and the running-count of elements greater-than-or-equal-to A[1] has been kept; since the sub-array is the entire array, a count of all elements greater-than-or-equal-to A[1] has been kept. Thus, the algorithm is correct as it keeps a count of all elements greater-than-or-equal-to A[1] for a given array.

4. Below is the Decision Tree for problem 4