CS-1203 – Monsoon 2023 – Assignment 1

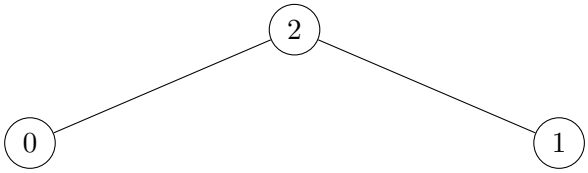
Ayush Kishor,

September 26, 2023

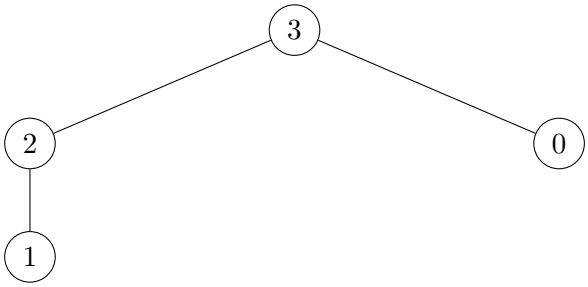
Heap Insertion Analysis

Comparison Counter = 1
Swap Counter = 0

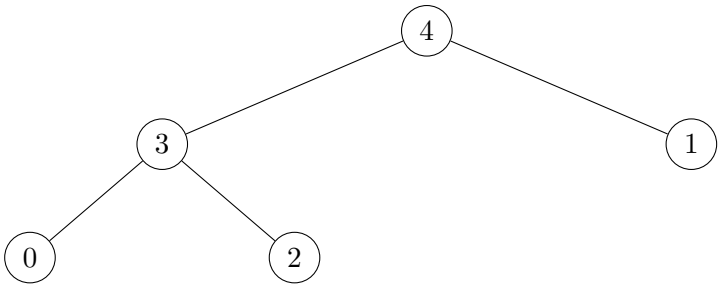
Comparison Counter = 2
Swap Counter = 1



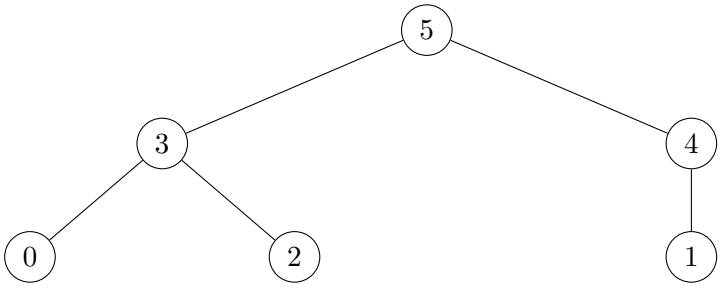
Comparison Counter = 3
Swap Counter = 2

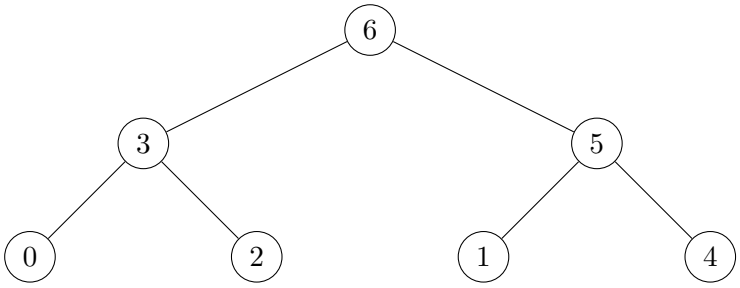


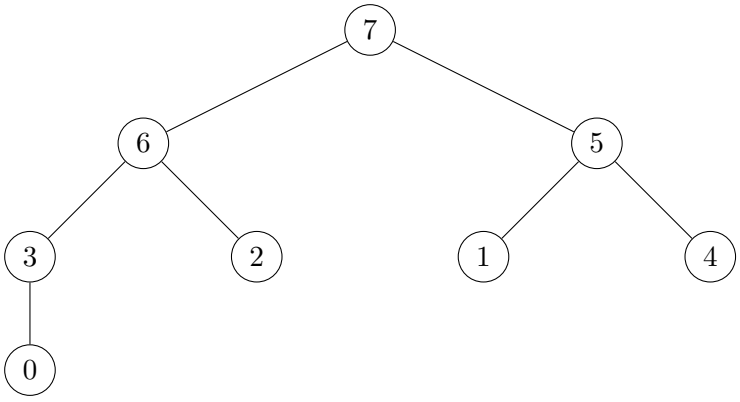
Comparison Counter = 5
Swap Counter = 4

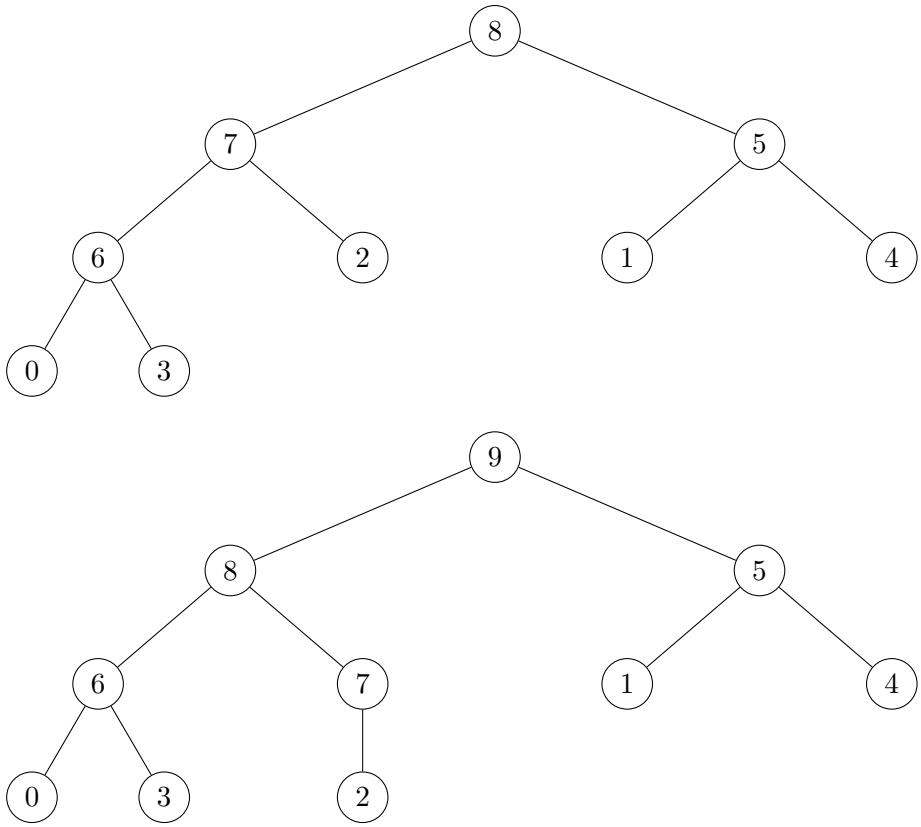


Comparison Counter = 7
Swap Counter = 6
and so on.....





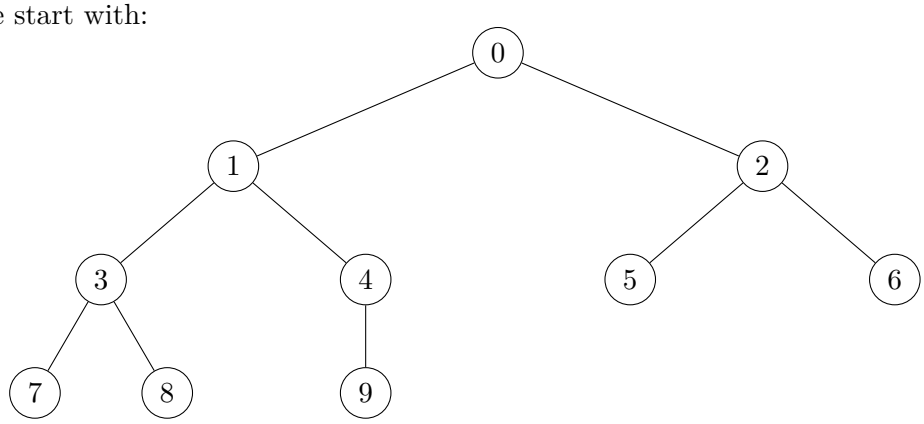




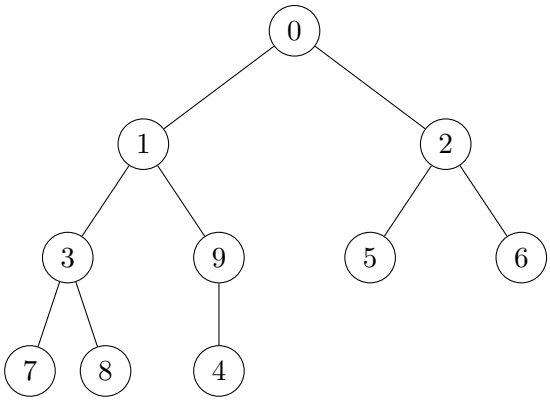
Now, we see a Max Heap has been formed through Heap Insert. Analysing the algorithm, we see that the number of operations(counts+swaps) = 39 which is roughly $O(nlog(n)$. Now let's analyse Heapify Algorithm

Heapify Analysis

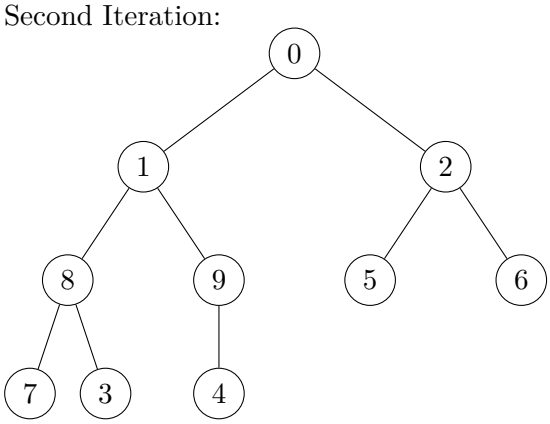
We start with:



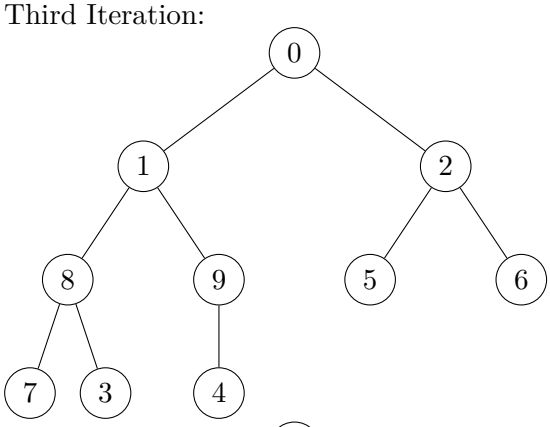
Now in our First Iteration of the For Loop, we have

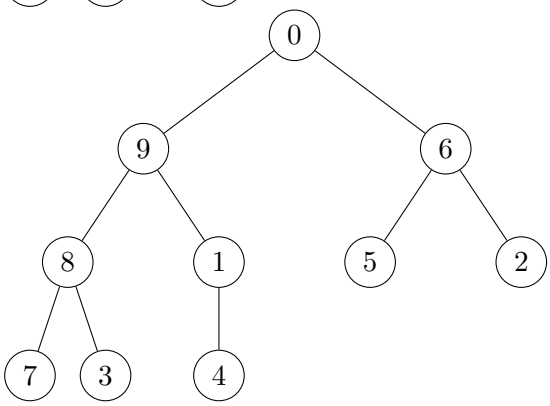


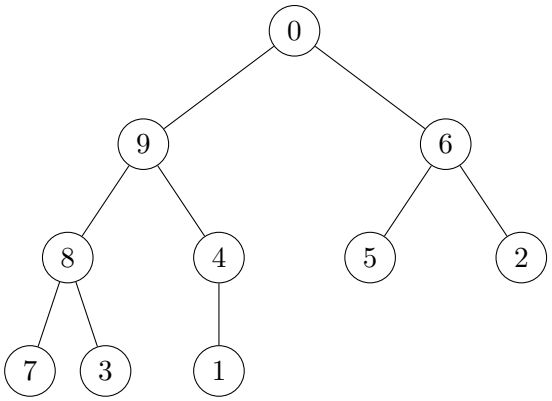
Second Iteration:



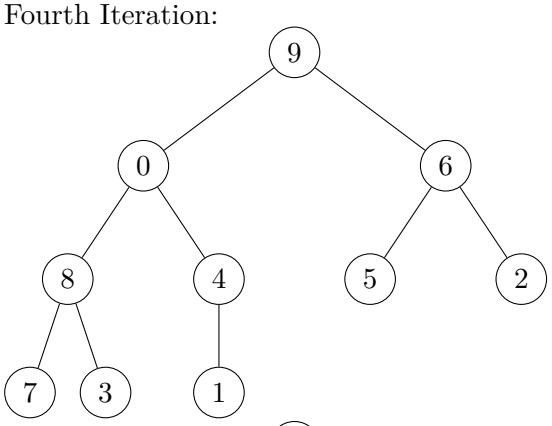
Third Iteration:

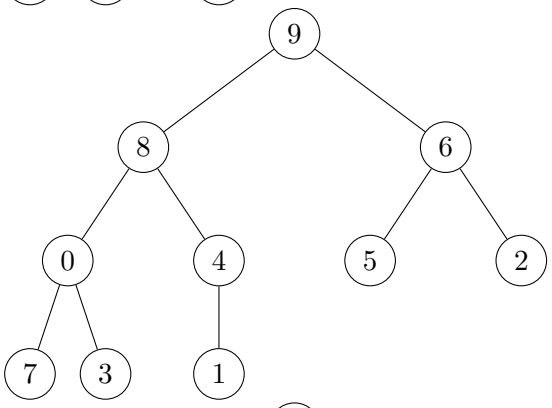


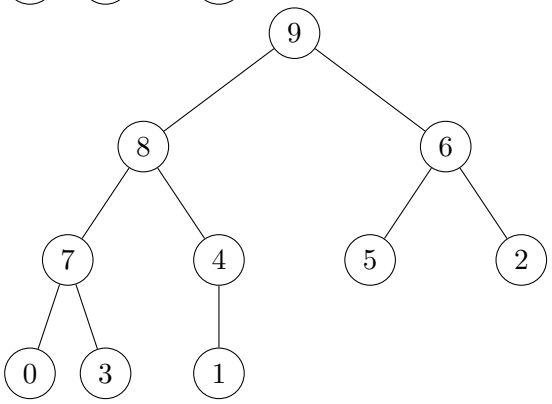




Fourth Iteration:







Counting the swaps and the comparisons, we get the total number of operations(comparisons + swaps) to be 28, which is roughly $3n$ or $O(n)$

Iteration	Comparison Count	Swap Count
1	3	1
2	6	2
3	$6+3+3$	$2+1+1$
4	$12+3+3+3$	$4+1+1+1$
Total	21	7

Heap Insert vs Heapify

From the analysis of both algorithms on an array of integers from 0 to 9, we observe the total number of operations vary by a considerable margin, especially swaps. We can ascertain the time complexities to be $O(nlog(n))$ and $O(n)$ respectively. Further, both algorithms come up with the same element i.e, a Max Heap but the ordering of elements vary.