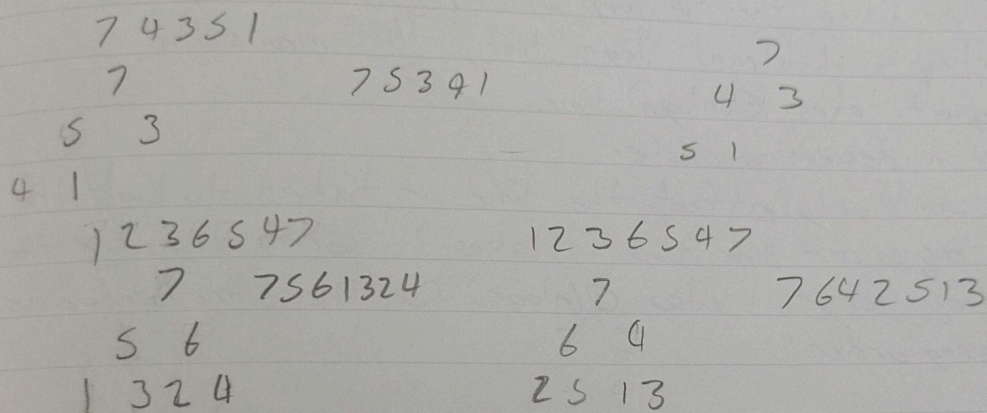


1) Build Max Heap (A)
 A.heapSize = 1
 for $i = 2$ to A.length
 MaxHeapInsert(A, A[i])



a) They do not result in the same heap
 The array [1 2 3 6 5 4 7] results in [7 5 6 1 3 2 4] done this way, & [7 6 4 2 5 1 3] done the way in the textbook

b) The new BuildMaxHeap requires $\Theta(n \log n)$ time for its worst case because the for loop runs n times & MaxHeapInsert takes $\log n$ times because it can be called once for each level of the tree if the inserted node is larger than all other nodes in the tree. This means that a $\log n$ function is called n times, or $\Theta(n \log n)$ if the initial array is already sorted in increasing order