Heap

Ali Akbari 30171539

April 2024

Creation of Max Heap and Heap Sort Time Complexity

For the creation of a max heap using Heapify the time complexity is O(n). This is because you go through each element in the list ensuring it is a max heap. This time complexity holds for both the average and the worst case of the algorithm. This is because in the worst case you still have to traverse through n elements. For the creation of a max heap using one by one it will be O($nlogn$). If we insert elements one by one into an initially empty heap to create a max-heap, we need to perform n insertions. Each insertion may require traversing the height of the heap, which is $logn$ in the worst case. Therefore, the time complexity for creating a max-heap using one-by-one insertion is O($nlogn$) in the worst case. The time complexity for the average case is still O($nlogn$). In the average case we still have to traverse the height of the heap which is O($logn$) and we still have to insert n elements which is O(n) Therefore making the average case O($nlogn$). The heap sort algorithm will have to make a max heap and then sort it. To make a max heap is O(n) using heapify. In the Heap Sort algorithm, after creating the max-heap, we repeatedly extract the maximum element from the heap and then re-heapify the remaining elements. Both operations take O($logn$) time. Since we perform these operations n times (once for each element), the overall time complexity for the sorting step is O($nlogn$) in both the average and worst cases.