Computational Complexity in Heap Sort

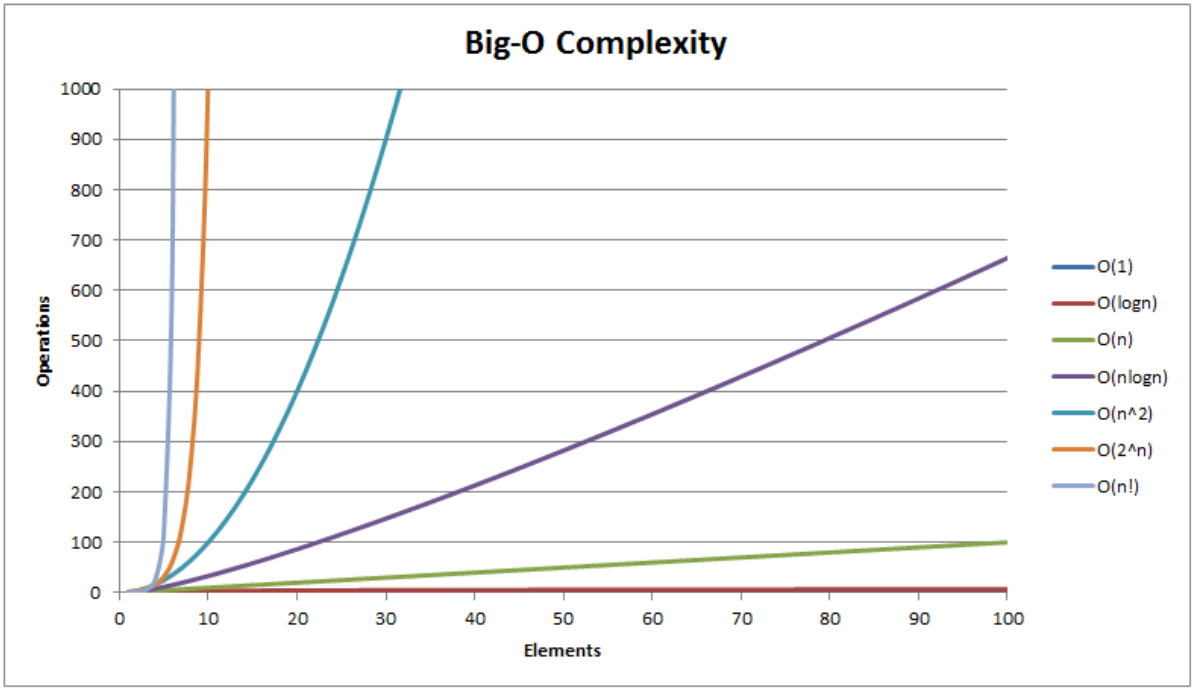
Time

The time complexity for best, worst and average case scenario is O(n*log n). In the context of other sorting algorithms, this means that it's faster than insertion, selection and bubble sort, even in the worst-case scenario. It has the same time complexity of a merge sort under any scenario and is better than a quick sort under a worst-case scenario.

Space

In terms of space complexity, heap sort has 0(1) in the worst case scenario. It's the same space complexity of bubble sort, selection sort and insertion sort. This means that the growth of memory is constant and does not depend on the size of the input. The reason it has a space complexity of 0(1) is because the heap is build inside the array to be sorted, if this didn't happen, the space complexity would be 0(n).

Sorting Algorithms	Time Complexity			Space Complexity
	Best Case	Average Case	Worst Case	Worst Case
Bubble Sort	O(n)	O(n^2)	O(n^2)	O(1)
Selection Sort	O(n^2)	O(n^2)	O(n^2)	O(1)
Insertion Sort	O(n)	O(n^2)	O(n^2)	O(1)
Merge Sort	O(nlogn)	O(nlogn)	O(nlogn)	O(n)
Quick Sort	O(nlogn)	O(nlogn)	O(n^2)	O(n)
Heap Sort	O(nlogn)	O(nlogn)	O(nlogn)	O(1)
Counting Sort	O(n + k)	O(n + k)	O(n + k)	O(k)
Radix Sort	O(nk)	O(nk)	O(nk)	O(n + k)
Bucket Sort	O(n + k)	O(n + k)	O(n^2)	O(n)



Sources:

https://www.geeksforgeeks.org/heap-sort/

https://www.interviewcake.com/concept/java/heapsort

https://www.programiz.com/dsa/heap-sort

Images:

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.hackerearth.com%2Fpractice%2Fnotes%2Fbig-o-cheatsheet-series-data-structures-and-algorithms-with-thier-complexities-1%2F&psig=AOvVaw3VQmeFXhnrkEJC13KdaOOf&ust=1652113402671000&source=images&cd=vfe&ved=0CA0QjhxqFwoTCJC646Co0PcCFQAAAAAdAAAAABAD

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.enjoyalgorithms.com%2Fblog%2Fcomparison-of-sorting-algorithms%2F&psig=AOvVaw2FCL-R7ZkWW4-FIKMjEG7N&ust=1652113570309000&source=images&cd=vfe&ved=0CA0QjhxqFwoTCKDz7vKo0PcCFQAAAAAdAAAAABAD