

Granit Gashi

Professor Dong

CSIT 212

February 9 2018

The time complexity of these 3 algorithms is kind of similar. They are all very very fast and complete the task near instant. The time complexity for a heap sort algorithms is  $O(n \log n)$ , and this is definitely faster than the last algorithms and you could actually see the difference on the execution. In reality the best sorting algorithm is Quick sort because it does not need to meet a special condition in order to be the best. Quicksort does its job the quickest and that is why it is named that. Time complexity is still  $O(n \log n)$  but it seems faster than the Heap sort and that is because there is less operations to do. The final sorting algorithms we learned in class was Counting which has a time complexity of  $(k+n)$  where  $k$  is the largest value in an array. Counting sort is very fast if the largest number is smaller than the array size then in theory this is the best one because it literally takes a second to do. As my Sort2Chart represents counting is the fastest but if and only if it meets its condition and then quicksort is the second best and 3rd is heap sorting.