1. Heap Sort:
2. Million Random:

Start: 22:56:41.724

End: 22:56:42.251

Time Taken: 0.527 sec

1. 10 Million Random:

Start: 22:58:25.534

End: 22:58:34.942

Time Taken: 9.408 sec

Algorithm Description: Heap sort is a comparison-based algorithm that is a lot similar to selection sort(but a lot better) in that it finds the largest number in the array and places it at the end. It does this heapifying the array, that is to say it sorts the array by making it into a binary tree called a heap. All nodes in the heap are either greater than or equal to each of its children. Once the heap is complete it takes the element in the top of the heap (which if heaped correctly would be the biggest number) and swaps it with the last element in the array the then that number is removed from the heap. The heap is reconstructed after each removal. Time complexity of the “heapify” is big O(Log n) but you have to do this n times so the time complexity is Big O (nLog n) in all cases. Compared with Quick Sort, Heap sort has a better worst case Big O(n Log n). Quick Sort has complexity Big O(n^2) for worst case. But in other cases, Quick Sort is fast.

1. Bin/Radix Sort:
2. Million Random:

Start: 22:59:58.340

End: 22:59:58.923

Time Taken: 0.583 sec

1. 10 Million Random:

Start: 23:02:43.329

End: 23:02:51.859

Time Taken: 8.53 sec

Algorithm Description: Bin or Radix sort is an integer sorting algorithm that sorts data with integer keys by grouping the keys by individual digits that share the same significant position and value (place value). Because integers can be used to represent strings (by assigning integer values to characters), radix sort works on data types other than just integers. This works by creating and distributing elements into buckets according to their radix. Bin sort uses both dynamic and static memory, dynamic memory to create NodeArray and static memory to creates “bins” thus it uses more memory the most sorts. The worst best and average case is Big O (nw) n being the length of the list and w being the width of the largest element in the array.