COEN 352 Assignment 2

Nicholas Gattuso 40007087

1. a – d) Answers given in .java files

e) These are the obtained run times:

For RandomGen with QSNormal

- 1 million inputs: 66925000 microseconds

- 10 thousand inputs: 77000 microseconds

- 1 hundred inputs: 17000 microseconds

- ten inputs: 16000 microseconds

For RandomGen with QSInsertion

- 1 million inputs: 19471000 microseconds

- 10 thousand inputs: 3036900 microseconds

For FixedGen with QSNormal

- 1 million inputs: 32491000 microseconds

- 10 thousand inputs: 78000 microseconds

1. Quicksort and insertion differ in performance through the execution time. Insertion on average takes ϴ(n2) and can be even faster when the list is nearly sorted. Quicksort on average takes ϴ(nlogn) and at worst ϴ(n2). ϴ(n2) is faster for small values of n. For this reason, we use diversion to insertion in which we optimize our performance time by selection a size to use insertion instead of quicksort. To obtain the size, I ran the program multiple times changing the diversion criteria until I saw a significant difference in time.

1. A fixedgen will create a pathological case because the list is already sorted but will still go through the sorting. Similarly, we get a worst case when the list is sorted in reverse order or all elements are the same. A badly chosen pivot can also give a worst case. I chose the pivot as middle of bottom and top because it is simple to arrange the numbers less than or greater. A pivot at either end of the array will give a worst case because it will compare only one value then recurse.