1. Implement both versions of quicksort (random and non-random choice for the pivot) -

Answer is in Quicksort (Random and Non-random) program along with output.

2. For the non-random pivot version of quicksort show the following benchmarks on the same graph:

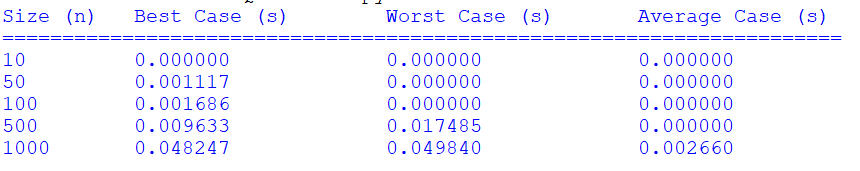
2a) best case (generate a set of inputs that will always be the best case, repeat for multiple array input sizes "n").

2b) worst case (generate a set of inputs that will always be the worst case, repeat for multiple array input sizes "n").

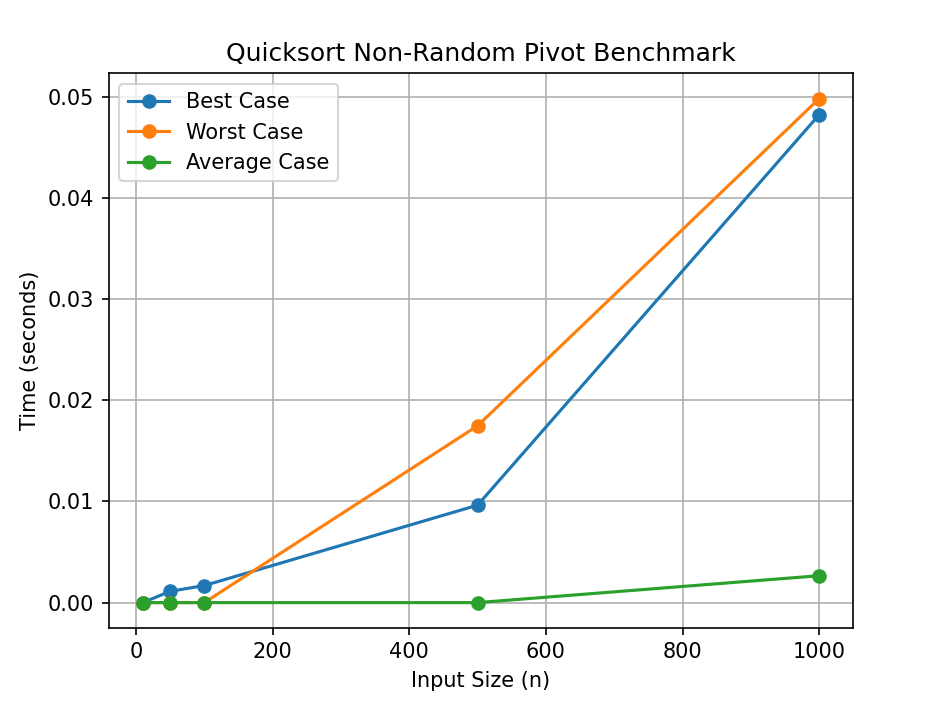
2c) average case (generate a set of inputs from a uniform distribution, repeat for multiple array input sizes "n").

Answer is in benchmark of non-random quicksort and the output and the benchmark is below

**Data plot:**



**Benchmarking:**



3. Mathematically derive the average runtime complexity of the non-random pivot version of quicksort.

