PROJECT 3 - Zebo Xiong | A04907051

QUICK SORT and ALGORITHM ANALYSIS

Using Quick Sort algorithms implement a Program in any language you desire

(C++ or java or Python) to sort an array of real numbers of size N.

Do both part A, B, and C.

A.

1. Input at least 4 or more sets of sorted data with at least N real numbers in each. For example, N= 10, 25, 35, and 45

```java
import java.util.Random;

public class TestRandom {

    public static void main(String[] args) {

        int[] all = {10, 25, 35, 45, 60, 70, 85, 95};

        for(int a : all) {

            System.out.print("N = " + a + "  -  ");

            for (int i = 0; i < a; i++)     System.out.print(getR(5, 70) + " ");

            System.out.println("");
        }
    }

    private static int getR(int min, int max) {

        Random r = new Random();
        return r.nextInt((max - min) + 1) + min;
    }

}
```

```
N = 10  -  10 37 53 18 42 34 48 56 36 38
Sorted - 10 18 34 36 37 38 42 48 53 56

N = 25  -  58 43 15 19 32 14 59 69 68 39 34 55 64 26 61 55 52 9 30 39 24 36 23 56 53
Sorted - 9 14 15 19 23 24 26 30 32 34 36 39 39 43 52 53 55 55 56 58 59 61 64 68 69

N = 35  -  10 6 69 8 5 44 21 53 5 70 70 19 49 15 53 68 18 42 24 36 50 62 58 16 27 17 14 12 33 52 12 69 62 16 63
Sorted - 5 5 6 8 10 12 12 14 15 16 16 17 18 19 21 24 27 33 36 42 44 49 50 52 53 53 58 62 62 63 68 69 69 70 70

N = 45  -  63 12 16 22 64 70 17 65 48 38 58 58 24 16 44 42 56 59 26 5 36 46 22 16 28 18 14 17 18 19 44 23 52 5 66 66 70 58 16 29 38 38 53 11 9
Sorted - 5 5 9 11 12 14 16 16 16 16 17 17 18 18 19 22 22 23 24 26 28 29 36 38 38 38 42 44 44 46 48 52 53 56 58 58 58 59 63 64 65 66 66 70 70

N = 60  -  11 53 44 48 14 43 20 29 46 13 29 36 66 52 58 51 27 28 25 16 35 49 35 28 50 30 27 38 62 24 11 53 24 53 49 53 45 16 7 35 15 36 40 55 51 52
Sorted - 7 10 11 11 11 12 13 14 15 16 16 20 21 24 24 25 26 27 27 27 28 28 29 29 30 35 35 35 36 36 38 38 40 43 43 43 44 45 46 48 49 49 49 50 51 51 52

N = 70  -  35 38 25 63 60 34 62 25 25 14 9 28 10 24 25 70 18 11 8 57 46 52 67 50 11 41 53 48 56 51 27 12 37 12 12 40 33 44 6 25 5 10 67 9 33 8 41 47
Sorted - 5 6 6 8 8 9 9 10 10 11 11 12 12 12 14 17 18 20 20 21 24 24 25 25 25 25 25 26 27 28 30 33 33 34 35 37 38 39 40 40 41 41 41 42 42 44 45 46 47

N = 85  -  36 21 64 68 55 18 5 53 17 20 57 9 38 19 14 18 30 46 19 13 5 26 48 15 27 5 63 68 14 24 31 16 29 29 36 50 59 46 69 24 32 19 58 47 64 42 54
Sorted - 5 5 5 9 9 11 13 14 14 14 15 16 16 17 18 18 18 18 19 19 19 20 21 21 23 24 24 25 26 26 27 27 28 28 29 29 29 29 29 30 30 31 31 32 36 36 37 38

N = 95  -  30 39 59 70 39 10 46 9 69 24 18 37 49 43 30 44 38 8 54 62 39 51 23 57 21 29 27 54 48 10 55 26 13 31 70 6 66 15 46 32 45 42 59 69 49 7 49
Sorted - 6 7 7 8 8 8 9 9 9 10 10 10 12 12 13 13 14 15 15 17 17 18 18 19 21 21 23 24 26 27 29 30 30 31 31 32 33 34 35 37 38 38 39 39 39 39 40 40 41 4
```

2. Plot a graph to compare the Worst Case Complexity of the quick sort algorithm and actual count putting counters in strategic points of your programs. (Time counter not allowed.)
   Input data must be good for Worst Case Quick Sort.

N = 10 - 67 28 5 40 42 13 49 19 66 15
Sorted - 5 13 15 19 28 40 42 49 66 67
Worst Case: N^2 = 100
Average Case: N^LogN = 30
Actual Count : = 54

N = 25 - 50 56 57 64 70 15 16 34 6 32 60 68 65 68 36 30 7 40 39 18 29 34 8 38 33
Sorted - 6 7 8 15 16 18 29 30 32 33 34 34 36 38 39 40 50 56 57 60 64 65 68 68 70
Worst Case: N^2 = 625
Average Case: N^LogN = 100
Actual Count : = 324

N = 35 - 8 42 24 7 52 54 49 17 53 18 54 64 38 29 63 48 32 55 38 21 31 47 63 13 37 49 47 29 59 31 44 40 17 43 61
Sorted - 7 8 13 17 17 18 21 24 29 29 31 31 32 37 38 38 40 42 43 44 47 47 48 49 49 52 53 54 54 55 59 61 63 63 64
Worst Case: N^2 = 1225
Average Case: N^LogN = 175
Actual Count : = 629

N = 45 - 14 24 31 33 33 8 64 10 53 47 68 48 59 64 22 9 33 41 49 60 56 46 11 14 56 11 70 24 47 48 64 36 13 29 26 35 30 18 44 62 61 43 44 51 27
Sorted - 8 9 10 11 11 13 14 14 18 22 24 24 26 27 29 30 31 33 33 33 35 36 41 43 44 44 46 47 47 48 48 49 51 53 56 56 59 60 61 62 64 64 64 68 70
Worst Case: N^2 = 2025
Average Case: N^LogN = 225
Actual Count : = 1034

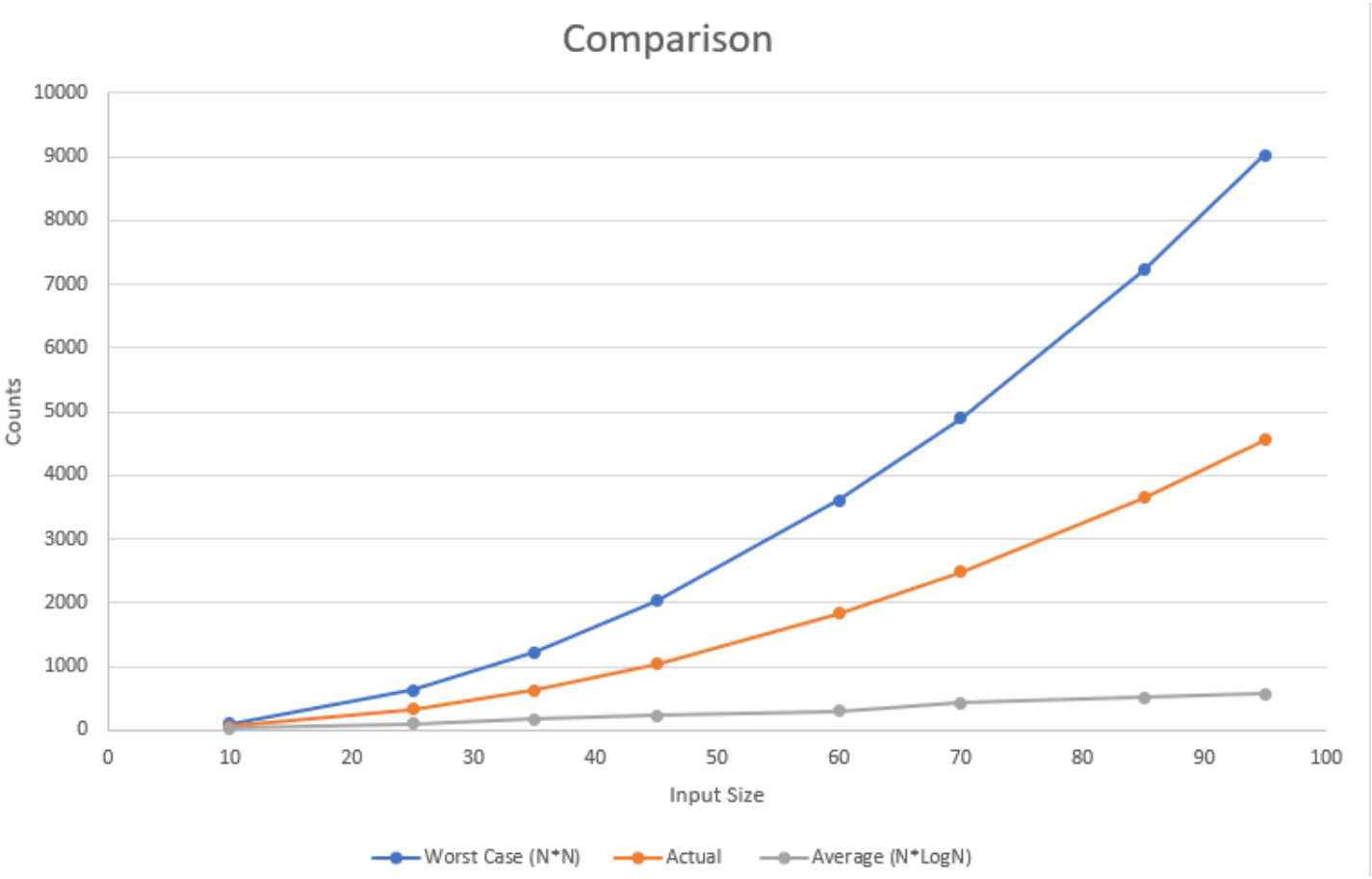
N = 60 - 26 57 62 6 61 57 53 37 43 66 9 14 11 22 40 24 62 50 44 30 55 38 67 57 35 16 70 13 36 14 50 54 30 9 56 55 35 42 5 51 31 54 15 45 58 18 61 69 11 7 58 19 60 19 50 32 19 16 54 10
Sorted - 5 6 7 9 9 10 11 11 13 14 14 15 16 16 18 19 19 19 22 24 26 30 30 31 32 35 35 36 37 38 40 42 43 44 45 50 50 50 51 53 54 54 54 55 55 56 57 57 57 58 58 60 61 61 62 62 66 67 69 70
Worst Case: N^2 = 3600
Average Case: N^LogN = 300
Actual Count : = 1829

N = 70 - 63 40 27 66 36 21 25 22 34 13 37 56 65 30 30 40 10 24 17 35 41 62 55 55 13 10 32 60 24 39 11 69 47 24 29 34 22 42 31 13 27 12 66 67 69 9 59 33 49 62 14 12 41 33 18 32 12 34 54 48 14 16 21 63 18 29 43 50 29 56
Sorted - 9 10 10 11 12 12 12 13 13 13 14 14 16 17 18 18 21 21 22 22 24 24 24 25 27 27 29 29 29 30 30 31 32 32 33 33 34 34 34 35 36 37 39 40 40 41 41 42 43 47 48 49 50 54 55 55 56 56 59 60 62 62 63 63 65 66 66 67 69 69
Worst Case: N^2 = 4900
Average Case: N^LogN = 420
Actual Count : = 2484

N = 85 - 41 13 61 8 22 11 55 60 66 42 44 65 12 7 58 10 10 46 22 38 54 54 65 64 6 69 69 50 37 54 31 48 64 52 54 68 57 11 60 62 18 48 27 18 50 59 70 8 9 22 22 49 35 50 44 6 28 8 32 46 40 21 51 64 58 48 44 44 10 20 20 51 44 70 22 40 61 37 62 25 15 58 67 14 30
Sorted - 6 6 7 8 8 8 9 10 10 10 11 11 12 13 14 15 18 18 20 20 21 22 22 22 22 22 25 27 28 30 31 32 35 37 37 38 40 40 41 42 44 44 44 44 44 46 46 48 48 48 49 50 50 50 51 51 52 54 54 54 54 55 57 58 58 58 59 60 60 61 61 62 62 64 64 64 65 65 66 67 68 69 69 70 70
Worst Case: N^2 = 7225
Average Case: N^LogN = 510

Actual Count : = 3654

N = 95 - 53 8 42 51 9 63 20 26 40 17 20 46 30 53 36 20 52 45 9 17 22 6 17 33 13 31 27 30 45 30 22 52 50 44 31 66 32 42 39 33 16 50 65 66 44 14 14 54 51 46 63 35 70 34 66 45 31 27 49 5 25 29 6 23 19 20 48 15 66 44 68 49 21 59 35 69 50 7 58 48 13 44 55 43 47 59 56 11 17 51 14 6 67 20 32
Sorted - 5 6 6 6 7 8 9 9 11 13 13 14 14 14 15 16 17 17 17 17 19 20 20 20 20 20 21 22 22 23 25 26 27 27 29 30 30 30 31 31 31 32 32 33 33 34 35 35 36 39 40 42 42 43 44 44 44 44 45 45 45 46 46 47 48 48 49 49 50 50 50 51 51 51 52 52 53 53 54 55 56 58 59 59 63 63 65 66 66 66 66 67 68 69 70
Worst Case: N^2 = 9025
Average Case: N^LogN = 570
Actual Count : = 4559



3. Display the beginning sorted data input and ending sorted data output for each array of N elements.

```
N = 10  -  10 37 53 18 42 34 48 56 36 38
Sorted - 10 18 34 36 37 38 42 48 53 56

N = 25  -  58 43 15 19 32 14 59 69 68 39 34 55 64 26 61 55 52 9 30 39 24 36 23 56 53
Sorted - 9 14 15 19 23 24 26 30 32 34 36 39 39 43 52 53 55 55 56 58 59 61 64 68 69

N = 35  -  10 6 69 8 5 44 21 53 5 70 70 19 49 15 53 68 18 42 24 36 50 62 58 16 27 17 14 12 33 52 12 69 62 16 63
Sorted - 5 5 6 8 10 12 12 14 15 16 16 17 18 19 21 24 27 33 36 42 44 49 50 52 53 53 58 62 62 63 68 69 69 70 70

N = 45  -  63 12 16 22 64 70 17 65 48 38 58 58 24 16 44 42 56 59 26 5 36 46 22 16 28 18 14 17 18 19 44 23 52 5 66 66 70 58 16 29 38 38 53 11 9
Sorted - 5 5 9 11 12 14 16 16 16 16 17 17 18 18 19 22 22 23 24 26 28 29 36 38 38 38 42 44 44 46 48 52 53 56 58 58 58 59 63 64 65 66 66 70 70

N = 60  -  11 53 44 48 14 43 20 29 46 13 29 36 66 52 58 51 27 28 25 16 35 49 35 28 50 30 27 38 62 24 11 53 24 53 49 53 45 16 7 35 15 36 40 55 51 52
Sorted - 7 10 11 11 11 12 13 14 15 16 16 20 21 24 24 25 26 27 27 27 28 28 29 29 30 35 35 35 36 36 38 38 40 43 43 43 44 45 46 48 49 49 49 50 51 51 52

N = 70  -  35 38 25 63 60 34 62 25 25 14 9 28 10 24 25 70 18 11 8 57 46 52 67 50 11 41 53 48 56 51 27 12 37 12 12 40 33 44 6 25 5 10 67 9 33 8 41 47
Sorted - 5 6 6 8 8 9 9 10 10 11 11 12 12 12 14 17 18 20 20 21 24 24 25 25 25 25 25 26 27 28 30 33 33 34 35 37 38 39 40 40 41 41 41 42 42 44 45 46 47

N = 85  -  36 21 64 68 55 18 5 53 17 20 57 9 38 19 14 18 30 46 19 13 5 26 48 15 27 5 63 68 14 24 31 16 29 29 36 50 59 46 69 24 32 19 58 47 64 42 54
Sorted - 5 5 5 9 9 11 13 14 14 14 15 16 16 17 18 18 18 18 19 19 19 20 21 21 23 24 24 25 26 26 27 27 28 28 29 29 29 29 29 30 30 31 31 32 36 36 37 38

N = 95  -  30 39 59 70 39 10 46 9 69 24 18 37 49 43 30 44 38 8 54 62 39 51 23 57 21 29 27 54 48 10 55 26 13 31 70 6 66 15 46 32 45 42 59 69 49 7 49
Sorted - 6 7 7 8 8 8 9 9 9 10 10 10 12 12 13 13 14 15 15 17 17 18 18 19 21 21 23 24 26 27 29 30 30 31 31 32 33 34 35 37 38 38 39 39 39 39 40 40 41 4
```

4. Display a table of N, Actual count and T(N)= Theta (n**2).

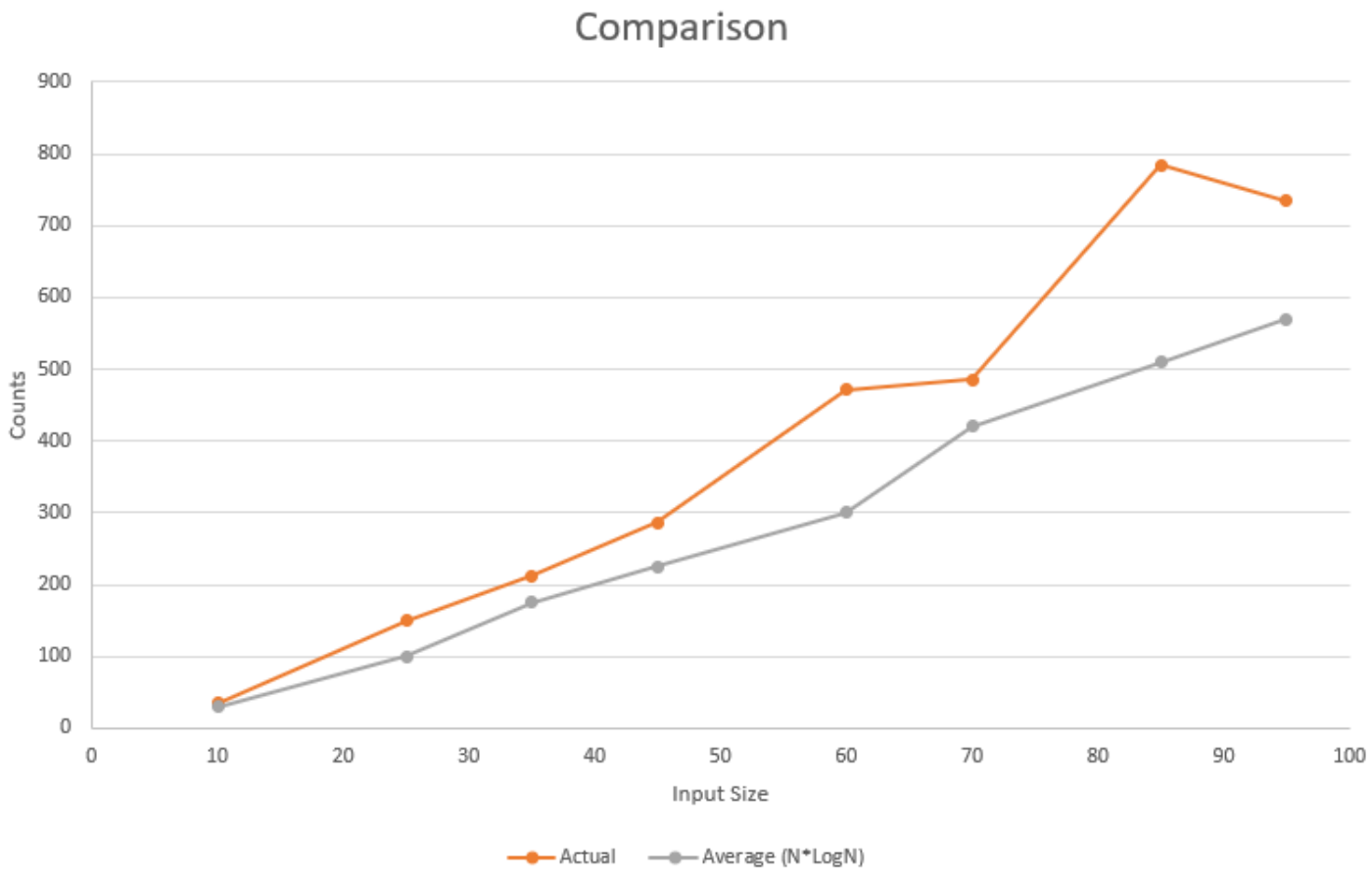
Input	Worst Case (N*N)	Average (N*LogN)	Actual
10	100	30	54
25	625	100	324
35	1225	175	629
45	2025	225	1034
60	3600	300	1829
70	4900	420	2484
85	7225	510	3654
95	9025	570	4559

B.

5. Input at least 4 or more sets of data with at least N real numbers in each set by using random generator for partition. For example, N= 10, 25, 35, and 45.

```
N = 10  -  50 48 34 22 47 5 19 45 15 32
Sorted  -  5 15 19 22 32 34 45 47 48 50
Worst Case: N^2 = 100
Average Case: N^LogN = 30
Actual Count : = 35

N = 25  -  5 34 32 24 58 68 18 6 37 65 14 58 30 24 51 47 32 24 27 6 37 42 34 60 59
Sorted  -  5 6 6 14 18 24 24 24 27 30 32 32 34 34 37 37 42 47 51 58 58 59 60 65 68
Worst Case: N^2 = 625
Average Case: N^LogN = 100
Actual Count : = 149

N = 35  -  64 11 21 26 16 54 42 9 37 5 64 33 17 27 42 21 25 30 55 25 8 16 39 38 48 69 25 7 7 53 46 29 33 47 12
Sorted  -  5 7 7 8 9 11 12 16 16 17 21 21 25 25 25 26 27 29 30 33 33 37 38 39 42 42 46 47 48 53 54 55 64 64 69
Worst Case: N^2 = 1225
Average Case: N^LogN = 175
Actual Count : = 212

N = 45  -  66 30 56 43 23 26 7 68 6 44 38 50 12 68 17 15 14 26 30 41 40 39 50 64 6 17 48 13 24 18 70 59 5 38 62 23 20 5 63 36 68 31 70 68 12
Sorted  -  5 5 6 6 7 12 12 13 14 15 17 17 18 20 23 23 24 26 26 30 30 31 36 38 38 39 40 41 43 44 48 50 50 56 59 62 63 64 66 68 68 68 70 70
Worst Case: N^2 = 2025
Average Case: N^LogN = 225
Actual Count : = 286

N = 60  -  11 59 38 45 30 10 26 36 17 67 6 15 21 54 56 37 34 60 50 22 41 47 11 46 29 7 42 68 62 17 67 50 6 44 5 43 7 54 32 36 11 70 36 30 11 28 47 27 52 63 36 39 25 41 35 6 62 17 46 18
Sorted  -  5 6 6 6 7 7 10 11 11 11 11 15 17 17 17 18 21 22 25 26 27 28 29 30 30 32 34 35 36 36 36 37 38 39 41 41 42 43 44 45 46 46 47 47 50 50 52 54 54 56 59 60 62 62 63 67 67 68 70
Worst Case: N^2 = 3600
Average Case: N^LogN = 300
Actual Count : = 471

N = 70  -  34 32 13 13 24 29 23 28 65 31 36 30 43 69 26 12 28 48 25 61 8 69 6 66 42 37 13 19 12 54 27 43 14 13 26 62 16 70 51 24 25 35 30 28 8 43 6 58 37 36 51 69 43 30 38 63 52 46 38 46 7 65
Sorted  -  6 6 7 8 8 12 12 13 13 13 13 14 16 19 23 24 24 25 25 25 26 26 27 27 28 28 28 28 29 30 30 30 31 32 34 34 35 36 36 37 37 38 38 42 43 43 43 43 46 46 48 48 51 51 52 54 55 58 61 62 63 65
Worst Case: N^2 = 4900
Average Case: N^LogN = 420
Actual Count : = 486

N = 85  -  64 10 61 41 60 48 33 11 39 55 14 48 16 43 31 47 59 34 50 46 5 45 37 44 36 65 9 55 57 58 57 13 56 59 15 57 43 62 14 53 51 6 32 29 56 68 5 12 70 45 67 53 24 10 42 13 35 25 28 56 6 68
Sorted  -  5 5 6 6 6 8 9 9 10 10 10 11 12 13 13 14 14 15 15 16 17 23 24 24 25 28 29 30 31 31 32 33 33 34 35 36 37 39 41 41 42 43 43 44 45 45 46 46 47 48 48 49 49 49 50 50 51 52 53 53 53 55 55
Worst Case: N^2 = 7225
Average Case: N^LogN = 510
Actual Count : = 784

N = 95  -  40 19 22 35 26 45 55 42 26 48 66 63 14 19 10 55 39 13 31 38 8 31 35 14 5 69 49 45 49 37 22 24 30 13 53 11 24 10 44 48 43 9 18 48 16 48 16 28 36 26 63 63 39 43 9 11 11 31 25 44 19 52
Sorted  -  5 5 6 7 7 8 9 9 9 10 10 10 11 11 11 11 12 13 13 14 14 15 15 16 16 17 18 19 19 19 19 19 22 22 23 24 24 24 25 26 26 26 28 30 31 31 31 31 32 32 34 35 35 36 37 38 39 39 40 41 42 4
Worst Case: N^2 = 9025
Average Case: N^LogN = 570
Actual Count : = 734
```

6. Plot a graph to compare the Complexity of the quick sort algorithm and actual count putting counters in strategic points of your programs.
   Input data must be good for general Quick Sort randomized algorithm.



7. Display the beginning data input and ending sorted data output for each array of N elements.

N = 10 - 50 48 34 22 47 5 19 45 15 32
Sorted - 5 15 19 22 32 34 45 47 48 50
Worst Case: N^2 = 100
Average Case: N^LogN = 30
Actual Count : = 35

N = 25 - 5 34 32 24 58 68 18 6 37 65 14 58 30 24 51 47 32 24 27 6 37 42 34 60 59
Sorted - 5 6 6 14 18 24 24 24 27 30 32 32 34 34 37 37 42 47 51 58 58 59 60 65 68
Worst Case: N^2 = 625
Average Case: N^LogN = 100
Actual Count : = 149

N = 35 - 64 11 21 26 16 54 42 9 37 5 64 33 17 27 42 21 25 30 55 25 8 16 39 38 48 69 25 7 7 53 46 29 33 47 12
Sorted - 5 7 7 8 9 11 12 16 16 17 21 21 25 25 25 26 27 29 30 33 33 37 38 39 42 42 46 47 48 53 54 55 64 64 69
Worst Case: N^2 = 1225
Average Case: N^LogN = 175
Actual Count : = 212

N = 45 - 66 30 56 43 23 26 7 68 6 44 38 50 12 68 17 15 14 26 30 41 40 39 50 64 6 17 48 13 24 18 70 59 5 38 62 23 20 5 63 36 68 31 70 68 12
Sorted - 5 5 6 6 7 12 12 13 14 15 17 17 18 20 23 23 24 26 26 30 30 31 36 38 38 39 40 41 43 44 48 50 50 56 59 62 63 64 66 68 68 68 68 70 70
Worst Case: N^2 = 2025
Average Case: N^LogN = 225
Actual Count : = 286

N = 60 - 11 59 38 45 30 10 26 36 17 67 6 15 21 54 56 37 34 60 50 22 41 47 11 46 29 7 42 68 62 17 67 50 6 44 5 43 7 54 32 36 11 70 36 30 11 28 47 27 52 63 36 39 25 41 35 6 62 17 46 18
Sorted - 5 6 6 6 7 7 10 11 11 11 11 15 17 17 17 18 21 22 25 26 27 28 29 30 30 32 34 35 36 36 36 36 37 38 39 41 41 42 43 44 45 46 46 47 47 50 50 52 54 54 56 59 60 62 62 63 67 67 68 70
Worst Case: N^2 = 3600
Average Case: N^LogN = 300
Actual Count : = 471

N = 70 - 34 32 13 13 24 29 23 28 65 31 36 30 43 69 26 12 28 48 25 61 8 69 6 66 42 37 13 19 12 54 27 43 14 13 26 62 16 70 51 24 25 35 30 28 8 43 6 58 37 36 51 69 43 30 38 63 52 46 38 46 7 65 27 70 48 55 28 65 25 34
Sorted - 6 6 7 8 8 12 12 13 13 13 13 14 16 19 23 24 24 25 25 25 26 26 27 27 28 28 28 28 29 30 30 30 31 32 34 34 35 36 36 37 37 38 38 42 43 43 43 43 46 46 48 48 51 51 52 54 55 58 61 62 63 65 65 65 66 69 69 69 70 70
Worst Case: N^2 = 4900
Average Case: N^LogN = 420
Actual Count : = 486

N = 85 - 64 10 61 41 60 48 33 11 39 55 14 48 16 43 31 47 59 34 50 46 5 45 37 44 36 65 9 55 57 58 57 13 56 59 15 57 43 62 14 53 51 6 32 29 56 68 5 12 70 45 67 53 24 10 42 13 35 25 28 56 6 68 49 46 49 49 23 24 30 52 50 15 33 41 9 55 64 8 17 31 53 70 6 10 59
Sorted - 5 5 6 6 6 8 9 9 10 10 10 11 12 13 13 14 14 15 15 16 17 23 24 24 25 28 29 30 31 31 32 33 33 34 35 36 37 39 41 41 42 43 43 44 45 45 46 46 47 48 48 49 49 49 50 50 51 52 53 53 53 55 55 55 56 56 56 57 57 57 58 59 59 59 60 61 62 64 64 65 67 68 68 70 70
Worst Case: N^2 = 7225
Average Case: N^LogN = 510
Actual Count : = 784

N = 95 - 40 19 22 35 26 45 55 42 26 48 66 63 14 19 10 55 39 13 31 38 8 31 35 14 5 69 49 45 49 37 22 24 30 13 53 11 24 10 44 48 43 9 18 48 16 48 16 28 36 26 63 63 39 43 9 11 11 31 25 44 19 57 32 57 31 61 11 6 7 45 56 10 5 7 9 26 15 15 31 19 46 44 12 34 23 41 32 19 49 17 64 24 66 44 52

Sorted - 5 5 6 7 7 8 9 9 9 10 10 10 11 11 11 11 12 13 13 14 14 15 15 16 16 17 18 19 19 19 19 19 22 22 23 24 24 24 25 26 26 26 26 28 30 31 31 31 31 31 32 32 34 35 35 36 37 38 39 39 40 41 42 43 43 44 44 44 44 45 45 45 46 48 48 48 48 49 49 49 52 53 55 55 56 57 57 61 63 63 63 64 66 66 69
Worst Case: N^2 = 9025
Average Case: N^LogN = 570
Actual Count : = 734

8. Display a table of N, Actual count and T(N)= Theta (n lg n).

Input	Average (N*LogN)	Actual
10	30	30
25	100	149
35	175	212
45	225	286
60	300	471
70	420	486
85	510	784
95	570	734

C.

9. Use the original version (1962) of Quick Sort by C A R Hoare which is listed in page 185, problem 7.1 of Cormen's book and do part B again

```
HOARE-PARTITION(A, p, r)
1   x = A[p]
2   i = p − 1
3   j = r + 1
4   while TRUE
5       repeat
6           j = j − 1
7       until A[j] ≤ x
8       repeat
9           i = i + 1
10      until A[i] ≥ x
11      if i < j
12          exchange A[i] with A[j]
13      else return j
```

```
180     /**  Hoare-Partition  **/
181     private static int partition(int[] A, int p, int r){
182
183         int x = A[p];
184
185         int i = p - 1;
186
187         int j = r + 1;
188
189         while(true){
190
191             do {
192                 j--;
193             } while ( A[j] > x);
194
195             do {
196                 i++;
197             } while ( A[i] < x);
198
199             if(i<j){
200
201                 int tmp = A[i];
202                 A[i] = A[j];
203                 A[j] = tmp;
204
205             } else return j;
206
207
208         }
209
210     }
```

N = 10 - 39 6 23 52 15 56 49 64 65 24
Sorted - 15 6 23 24 39 49 52 56 64 65

--> Worst Case: N^2 = 100
--> Average Case: N^LogN = 30
--> Actual Count : = 82

N = 25 - 47 19 28 47 59 59 19 43 34 11 11 40 32 46 26 11 31 65 49 5 57 44 56 28 32
Sorted - 5 11 11 11 19 19 26 28 28 31 32 32 34 40 43 44 46 47 47 49 56 57 59 59 65
--> Worst Case: N^2 = 625
--> Average Case: N^LogN = 100
--> Actual Count : = 316

N = 35 - 63 69 69 67 5 16 9 44 25 34 59 8 6 9 64 14 43 65 67 67 69 9 31 52 23 33 63 26 47 11 52 14 51 11 22
Sorted - 5 8 9 6 9 9 11 11 14 14 16 22 23 25 26 33 31 34 43 44 47 51 52 52 63 59 63 64 65 67 67 67 69 69 69
--> Worst Case: N^2 = 1225
--> Average Case: N^LogN = 175
--> Actual Count : = 551

N = 45 - 61 68 37 31 6 12 51 43 44 33 50 10 38 63 69 46 64 68 19 48 29 15 37 5 65 66 42 19 31 51 49 7 68 59 21 10 55 10 37 10 27 15 28 62 52
Sorted - 5 6 7 10 10 10 10 12 15 19 15 19 21 27 28 31 31 29 33 37 37 37 38 42 43 44 48 49 46 50 51 51 52 55 59 61 63 62 64 65 66 68 68 68 69
--> Worst Case: N^2 = 2025
--> Average Case: N^LogN = 225
--> Actual Count : = 802

N = 60 - 46 20 22 28 46 43 55 5 6 5 11 38 47 9 61 37 62 37 47 64 19 25 70 33 22 54 66 39 35 60 30 28 63 44 16 20 16 55 64 37 64 29 18 33 18 61 68 36 10 60 64 16 64 27 20 65 39 14 61 38
Sorted - 5 5 6 9 10 11 16 14 16 16 18 18 19 20 20 22 20 22 25 27 28 29 28 30 33 33 35 36 37 37 37 38 38 39 39 43 44 46 46 47 47 55 54 55 60 60 61 61 62 61 63 64 64 64 64 64 65 66 68 70
--> Worst Case: N^2 = 3600
--> Average Case: N^LogN = 300
--> Actual Count : = 1410

N = 70 - 30 60 69 60 7 41 37 60 66 21 57 7 64 43 24 64 64 14 15 61 62 37 68 59 61 53 49 34 48 19 47 40 10 37 24 33 58 46 29 19 14 17 60 42 68 44 68 34 15 37 49 37 9 59 53 54 46 65 55 9 49 52 16 18 66 48 43 53 21 16
Sorted - 7 9 7 10 14 9 14 15 15 16 16 17 19 18 19 21 21 24 24 29 30 33 34 34 37 37 37 37 37 40 42 43 41 43 44 46 46 48 47 48 49 49 49 52 53 53 53 55 54 57 58 59 59 60 60 60 60 61 61 62 64 64 64 65 66 66 68 68 68 69
--> Worst Case: N^2 = 4900
--> Average Case: N^LogN = 420
--> Actual Count : = 1792

N = 85 - 21 9 31 48 14 10 6 58 37 62 21 56 9 33 54 11 59 29 11 37 47 61 8 12 62 58 35 50 44 7 20 55 52 60 21 6 20 18 22 11 65 20 57 61 31 7 36 64 18 58 27 7 68 52 54 59 41 46 70 43 36 44 11 69 17 58 21 29 22 61 11 63 22 41 10 28 26 22 46 55 31 15 70 21 55
Sorted - 6 7 6 7 7 8 9 9 10 10 11 11 11 11 11 12 14 15 17 18 18 20 21 20 21 20 21 21 21 22 22 22 22 26 27 28 29 29 31 31 31 33 35 36 36 37 41 37 41 43 44 44 46 46 47 48 50 52 52 54 54 55 55 55 56 57 58 58 58 59 58 59 60 61 61 61 62 62 63 64 65 68 69 70 70
--> Worst Case: N^2 = 7225
--> Average Case: N^LogN = 510
--> Actual Count : = 2354

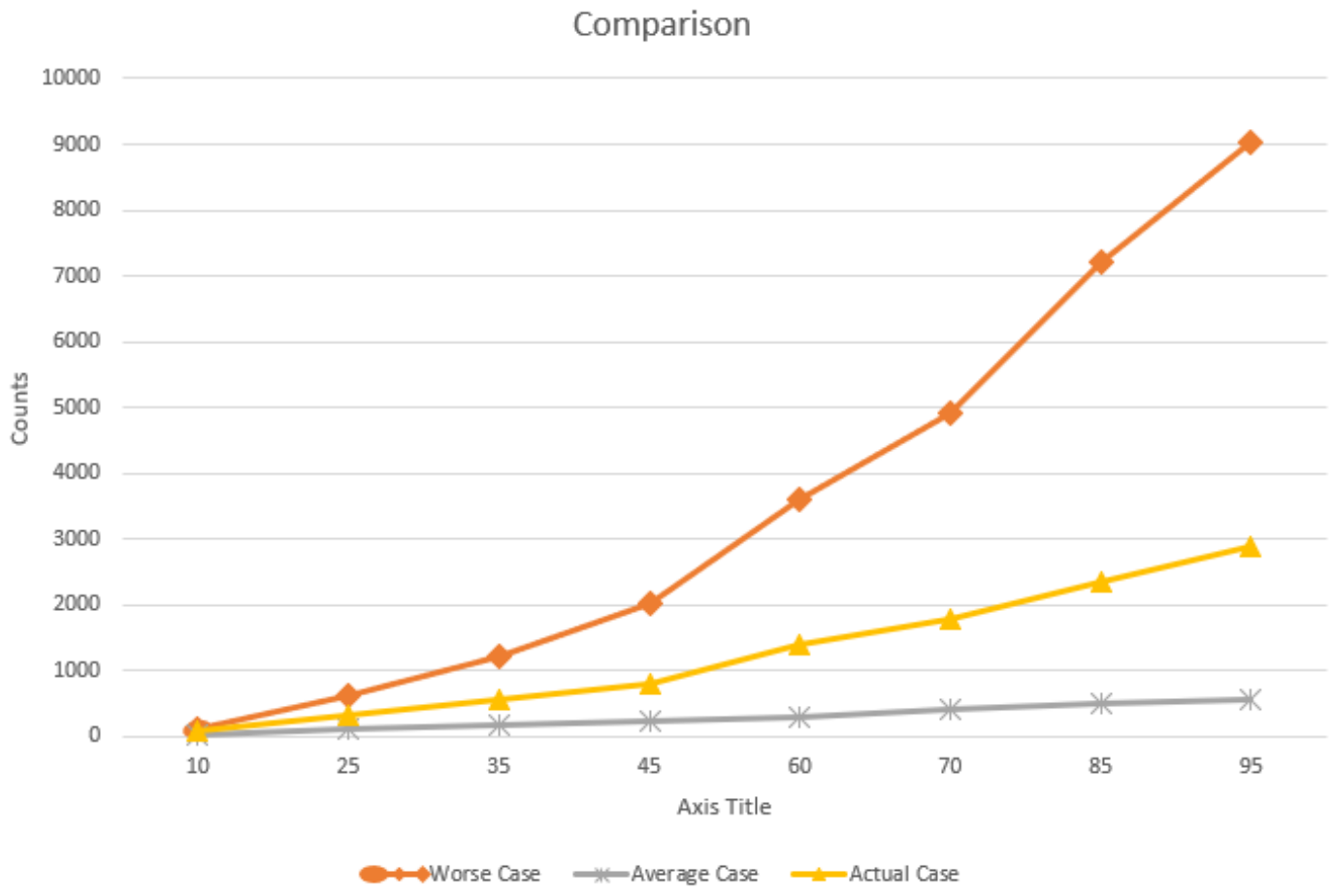
N = 95 - 34 6 58 35 11 30 43 43 26 50 15 18 8 20 68 61 59 15 35 58 65 66 27 26 41 64 60 45 12 20 32 53 23 64 66 37 6 38 43 6 61 36 6 25 47 58 57 44 31 28 35 34 39 11 53 13 51 21 9 60 21 5 8 21 52 30 9 54 23 24 40 46 16 42 27 67 36 63 36 12 48 63 15 42 35 64 7 8 17 46 17 29 43 66 68
Sorted - 5 6 6 6 6 7 8 8 8 9 11 9 11 12 12 13 15 15 15 16 17 17 18 20 20 21 21 21 23 23 24 25 26 26 27 27 28 29 30 30 31 34 32 34 35 35 35 36 35 36 36 37 38 39 40 41 42 42 43 43 43 43 45 46 44 46 47 48 50 51 53 52 53 54 57 58 58 58 59 60 60 61 61 63 63 64 64 64 66 65 66 66 68 67 68
--> Worst Case: N^2 = 9025
--> Average Case: N^LogN = 570

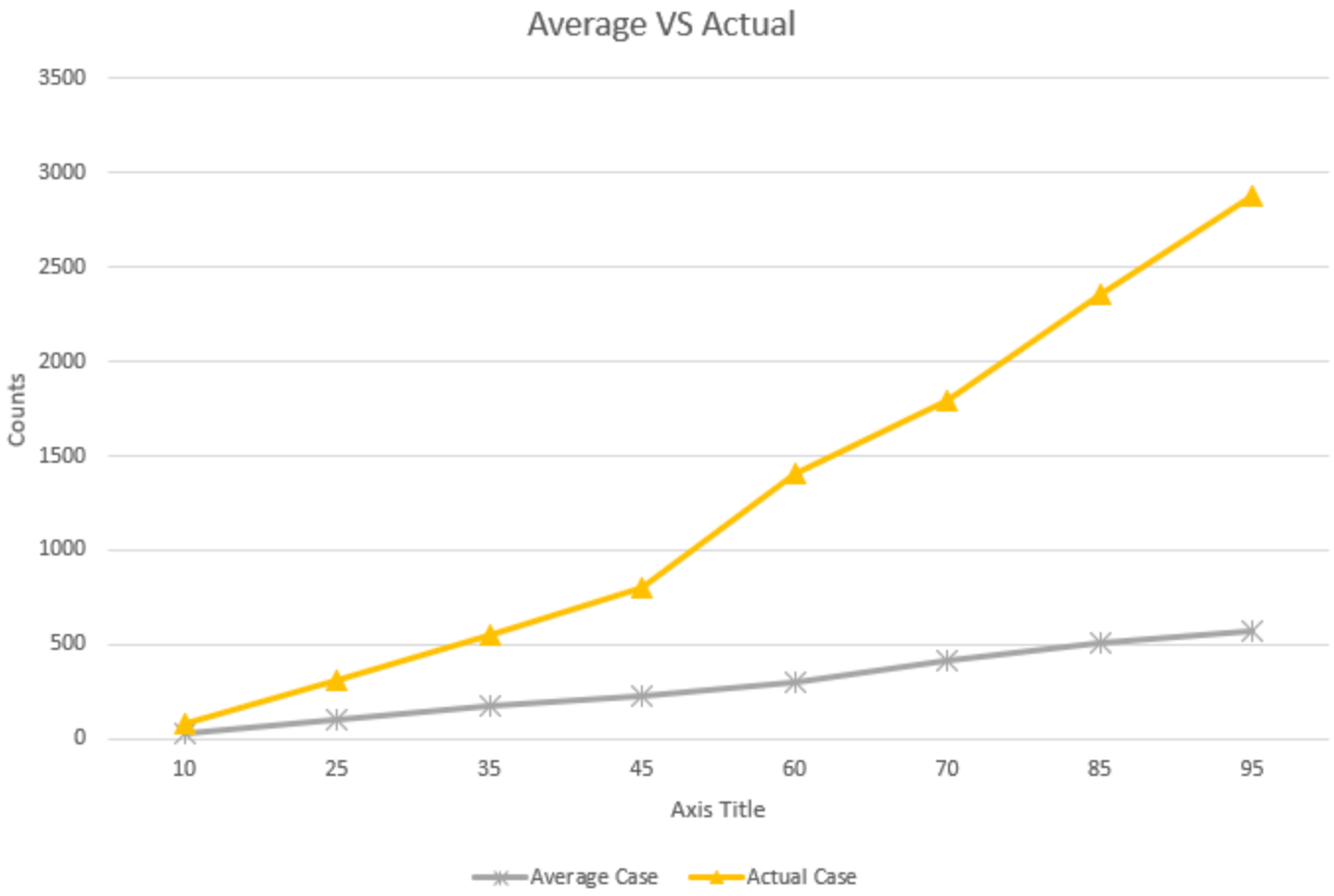
--> Actual Count : = 2877

10. The final graph of part C may be combined or separated from Part B.

```
N = 10  -  39 6 23 52 15 56 49 64 65 24
Sorted  -  15 6 23 24 39 49 52 56 64 65
--> Worst Case: N^2 = 100
--> Average Case: N^LogN = 30
--> Actual Count : = 82

N = 25  -  47 19 28 47 59 59 19 43 34 11 11 40 32 46 26 11 31 65 49 5 57 44 56 28 32
Sorted  -  5 11 11 11 19 19 26 28 28 31 32 32 34 40 43 44 46 47 47 49 56 57 59 59 65
--> Worst Case: N^2 = 625
--> Average Case: N^LogN = 100
--> Actual Count : = 316

N = 35  -  63 69 69 67 5 16 9 44 25 34 59 8 6 9 64 14 43 65 67 67 69 9 31 52 23 33 63 26 47 11 52 14 51 11 22
Sorted  -  5 8 9 6 9 9 11 11 14 14 16 22 23 25 26 33 31 34 43 44 47 51 52 52 63 59 63 64 65 67 67 67 69 69 69
--> Worst Case: N^2 = 1225
--> Average Case: N^LogN = 175
--> Actual Count : = 551

N = 45  -  61 68 37 31 6 12 51 43 44 33 50 10 38 63 69 46 64 68 19 48 29 15 37 5 65 66 42 19 31 51 49 7 68 59 21 10 55 10 37 10 27 15 28 62 52
Sorted  -  5 6 7 10 10 10 10 12 15 19 15 19 21 27 28 31 31 29 33 37 37 37 38 42 43 44 48 49 46 50 51 51 52 55 59 61 63 62 64 65 66 68 68 68 69
--> Worst Case: N^2 = 2025
--> Average Case: N^LogN = 225
--> Actual Count : = 802

N = 60  -  46 20 22 28 46 43 55 5 6 5 11 38 47 9 61 37 62 37 47 64 19 25 70 33 22 54 66 39 35 60 30 28 63 44 16 20 16 55 64 37 64 29 18 33 18 61 68 36 10 60 64
Sorted  -  5 5 6 9 10 11 16 14 16 16 18 18 19 20 20 22 20 22 25 27 28 29 28 30 33 33 35 36 37 37 37 38 38 39 39 43 44 46 46 47 47 55 54 55 60 60 61 61 62 61 63
--> Worst Case: N^2 = 3600
--> Average Case: N^LogN = 300
--> Actual Count : = 1410

N = 70  -  30 60 69 60 7 41 37 60 66 21 57 7 64 43 24 64 64 14 15 61 62 37 68 59 61 53 49 34 48 19 47 40 10 37 24 33 58 46 29 19 14 17 60 42 68 44 68 34 15 37 4
Sorted  -  7 9 7 10 14 9 14 15 15 16 16 17 19 18 19 21 21 24 24 29 30 33 34 34 37 37 37 37 37 40 42 43 41 43 44 46 46 48 47 48 49 49 49 52 53 53 53 55 54 57 58
--> Worst Case: N^2 = 4900
--> Average Case: N^LogN = 420
--> Actual Count : = 1792

N = 85  -  21 9 31 48 14 10 6 58 37 62 21 56 9 33 54 11 59 29 11 37 47 61 8 12 62 58 35 50 44 7 20 55 52 60 21 6 20 18 22 11 65 20 57 61 31 7 36 64 18 58 27 7 6
Sorted  -  6 7 6 7 7 8 9 9 10 10 11 11 11 11 11 12 14 15 17 18 18 20 21 20 21 20 21 21 21 22 22 22 22 26 27 28 29 29 31 31 31 33 35 36 36 37 41 37 41 43 44 44 4
--> Worst Case: N^2 = 7225
--> Average Case: N^LogN = 510
--> Actual Count : = 2354

N = 95  -  34 6 58 35 11 30 43 43 26 50 15 18 8 20 68 61 59 15 35 58 65 66 27 26 41 64 60 45 12 20 32 53 23 64 66 37 6 38 43 6 61 36 6 25 47 58 57 44 31 28 35 3
Sorted  -  5 6 6 6 6 7 8 8 8 9 11 9 11 12 12 13 15 15 15 16 17 17 18 20 20 21 21 21 23 23 24 25 26 26 27 27 28 29 30 30 31 34 32 34 35 35 35 36 35 36 36 37 38 3
--> Worst Case: N^2 = 9025
--> Average Case: N^LogN = 570
--> Actual Count : = 2877
```

Input Size	Worse Case	Average Case	Actual Case
10	100	30	82
25	625	100	316
35	1225	175	551
45	2025	225	802
60	3600	300	1410
70	4900	420	1792
85	7225	510	2354
95	9025	570	2877


Comparison
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
Counts
10
25
35
45
60
70
85
95
Axis Title
Worse Case
Average Case
Actual Case



11. Professional output with necessary information for readers are required.

    Done