# Programming Project

**Dwayne Fraser**

**Problem Definition**: The purpose of this project is to evaluate the run time complexity of three different algorithms. Each of the algorithms were given the input size of 10,000 - 200,000. The real-world applications of these algorithms dealt with the testing of sorting arrays and deciding the ith smallest element denoted as the order statistic.

**Algorithms and RT analysis:** The pseudocode for each algorithm and their run times are given below.

### Insertion Sort

```
INSERTION-SORT(A)
for j = 2 to A.length
    key = A[j]
    // Insert A[j] into the sorted sequence A[1 .. j − 1].
    i = j − 1
    while i > 0 and A[i] > key
        A[i + 1] = A[i]
        i = i − 1
    A[i + 1] = key
```

insertion sort has a
worst-case running time of ,$\Theta$(n^2)

### Heapsort

```
HEAPSORT(A)
BUILD-MAX-HEAP(A)
for i = A.length downto 2
    exchange A[1] with A[i]
    A.heap-size = A.heap-size − 1
    MAX-HEAPIFY(A, 1)
```

```
BUILD-MAX-HEAP(A)
A.heap-size = A.length
for i = ⌊A.length/2⌋ downto 1
    MAX-HEAPIFY(A, i)
```

```
MAX-HEAPIFY(A, i)
l = LEFT(i)
r = RIGHT(i)
if l ≤ A.heap-size and A[l] > A[i]
    largest = l
else largest = i
if r ≤ A.heap-size and A[r] > A[largest]
    largest = r
if largest ≠ i
    exchange A[i] with A[largest]
    MAX-HEAPIFY(A, largest)
```

The HEAPSORT procedure takes time O(n lg n)

## Randomized Select

```
RANDOMIZED-SELECT(A, p, r, i)
if p == r
    return A[p]
q = RANDOMIZED-PARTITION(A, p, r)
k = q − p + 1
if i == k          // the pivot value is the answer
    return A[q]
elseif i < k
    return RANDOMIZED-SELECT(A, p, q − 1, i)
else return RANDOMIZED-SELECT(A, q + 1, r, i − k)
```

```
RANDOMIZED-PARTITION(A, p, r)
i = RANDOM(p, r)
exchange A[r] with A[i]
return PARTITION(A, p, r)
```

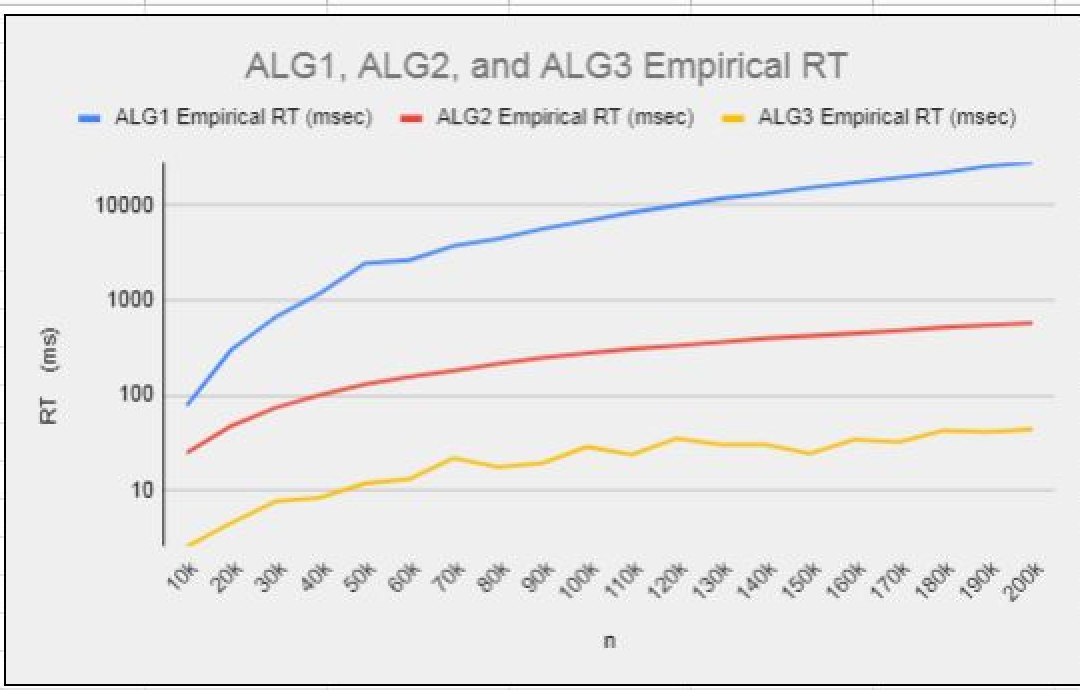
The new quicksort calls RANDOMIZED-PARTITION in place of PARTITION:

```
RANDOMIZED-QUICKSORT(A, p, r)
if p < r
    q = RANDOMIZED-PARTITION(A, p, r)
    RANDOMIZED-QUICKSORT(A, p, q − 1)
    RANDOMIZED-QUICKSORT(A, q + 1, r)
```

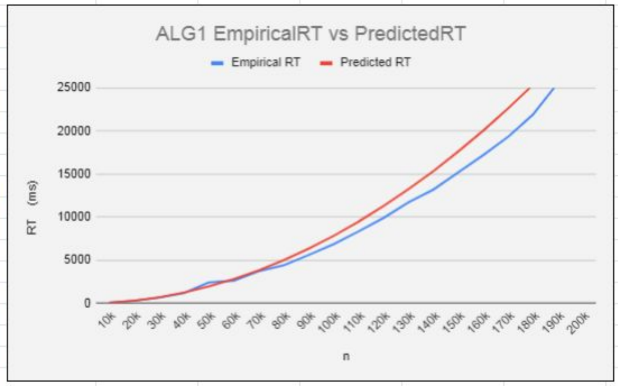
```
PARTITION(A, p, r)
x = A[r]
i = p − 1
for j = p to r − 1
    if A[j] ≤ x
        i = i + 1
        exchange A[i] with A[j]
exchange A[i + 1] with A[r]
return i + 1
```

The worst-case running time for RANDOMIZED-SELECT is $\Theta$(n^2)

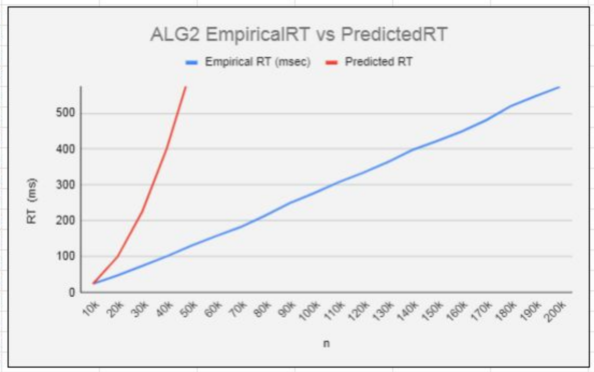
## Experimental Results:



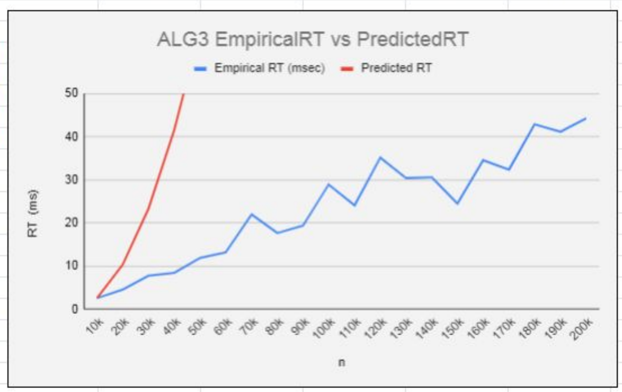
| N | Theoretical RT (n^2) | Empirical RT (msec) | Ratio (emp/theo) | Predicted RT |
|---|---|---|---|---|
| 1.00E+04 | 1.00E+08 | 78.362 | 7.84E-07 | 7.84E+01 |
| 2.00E+04 | 4.00E+08 | 302.937 | 7.57E-07 | 3.13E+02 |
| 3.00E+04 | 9.00E+08 | 672.71 | 7.47E-07 | 7.05E+02 |
| 4.00E+04 | 1.60E+09 | 1187.32 | 7.42E-07 | 1.25E+03 |
| 5.00E+04 | 2.50E+09 | 2435.01 | 9.74E-07 | 1.96E+03 |
| 6.00E+04 | 3.60E+09 | 2637.27 | 7.33E-07 | 2.82E+03 |
| 7.00E+04 | 4.90E+09 | 3726.09 | 7.60E-07 | 3.84E+03 |
| 8.00E+04 | 6.40E+09 | 4402.36 | 6.88E-07 | 5.02E+03 |
| 9.00E+04 | 8.10E+09 | 5615.2 | 6.93E-07 | 6.35E+03 |
| 1.00E+05 | 1.00E+10 | 6869.66 | 6.87E-07 | 7.84E+03 |
| 1.10E+05 | 1.21E+10 | 8353.79 | 6.90E-07 | 9.48E+03 |
| 1.20E+05 | 1.44E+10 | 9904.8 | 6.88E-07 | 1.13E+04 |
| 1.30E+05 | 1.69E+10 | 11729.1 | 6.94E-07 | 1.32E+04 |
| 1.40E+05 | 1.96E+10 | 13202.5 | 6.74E-07 | 1.54E+04 |
| 1.50E+05 | 2.25E+10 | 15203 | 6.76E-07 | 1.76E+04 |
| 1.60E+05 | 2.56E+10 | 17220.6 | 6.73E-07 | 2.01E+04 |
| 1.70E+05 | 2.89E+10 | 19346.6 | 6.69E-07 | 2.26E+04 |
| 1.80E+05 | 3.24E+10 | 21945.4 | 6.77E-07 | 2.54E+04 |
| 1.90E+05 | 3.61E+10 | 25589.8 | 7.09E-07 | 2.83E+04 |
| 2.00E+05 | 4.00E+10 | 27902.3 | 6.98E-07 | 3.13E+04 |




| n | N | Theoretical RT (n^2) | Empirical RT (msec) | Ratio (emp/theo) | Predicted RT |
|---|---|---|---|---|---|
| 10k | 1.00E+04 | 1.00E+08 | 25.0581 | 2.51E-07 | 2.51E+01 |
| 20k | 2.00E+04 | 4.00E+08 | 48.0206 | 1.20E-07 | 1.00E+02 |
| 30k | 3.00E+04 | 9.00E+08 | 74.5742 | 8.29E-08 | 2.26E+02 |
| 40k | 4.00E+04 | 1.60E+09 | 101.276 | 6.33E-08 | 4.01E+02 |
| 50k | 5.00E+04 | 2.50E+09 | 130.803 | 5.23E-08 | 6.26E+02 |
| 60k | 6.00E+04 | 3.60E+09 | 157.589 | 4.38E-08 | 9.02E+02 |
| 70k | 7.00E+04 | 4.90E+09 | 182.502 | 3.72E-08 | 1.23E+03 |
| 80k | 8.00E+04 | 6.40E+09 | 214.359 | 3.35E-08 | 1.60E+03 |
| 90k | 9.00E+04 | 8.10E+09 | 249.181 | 3.08E-08 | 2.03E+03 |
| 100k | 1.00E+05 | 1.00E+10 | 277.69 | 2.78E-08 | 2.51E+03 |
| 110k | 1.10E+05 | 1.21E+10 | 307.9 | 2.54E-08 | 3.03E+03 |
| 120k | 1.20E+05 | 1.44E+10 | 334.193 | 2.32E-08 | 3.61E+03 |
| 130k | 1.30E+05 | 1.69E+10 | 363.849 | 2.15E-08 | 4.23E+03 |
| 140k | 1.40E+05 | 1.96E+10 | 397.787 | 2.03E-08 | 4.91E+03 |
| 150k | 1.50E+05 | 2.25E+10 | 422.573 | 1.88E-08 | 5.64E+03 |
| 160k | 1.60E+05 | 2.56E+10 | 448.868 | 1.75E-08 | 6.41E+03 |
| 170k | 1.70E+05 | 2.89E+10 | 480.287 | 1.66E-08 | 7.24E+03 |
| 180k | 1.80E+05 | 3.24E+10 | 519.584 | 1.60E-08 | 8.12E+03 |
| 190k | 1.90E+05 | 3.61E+10 | 547.284 | 1.52E-08 | 9.05E+03 |
| 200k | 2.00E+05 | 4.00E+10 | 573.217 | 1.43E-08 | 1.00E+04 |




| N | Theoretical RT (n^2) | Empirical RT (msec) | Ratio (emp/theo) | Predicted RT |
|---|---|---|---|---|
| 1.00E+04 | 1.00E+08 | 2.59498 | 2.59E-08 | 2.59E+00 |
| 2.00E+04 | 4.00E+08 | 4.57073 | 1.14E-08 | 1.04E+01 |
| 3.00E+04 | 9.00E+08 | 7.76752 | 8.63E-09 | 2.34E+01 |
| 4.00E+04 | 1.60E+09 | 8.44184 | 5.28E-09 | 4.15E+01 |
| 5.00E+04 | 2.50E+09 | 11.9354 | 4.77E-09 | 6.49E+01 |
| 6.00E+04 | 3.60E+09 | 13.2023 | 3.67E-09 | 9.34E+01 |
| 7.00E+04 | 4.90E+09 | 21.9727 | 4.48E-09 | 1.27E+02 |
| 8.00E+04 | 6.40E+09 | 17.6744 | 2.76E-09 | 1.66E+02 |
| 9.00E+04 | 8.10E+09 | 19.4195 | 2.40E-09 | 2.10E+02 |
| 1.00E+05 | 1.00E+10 | 28.9085 | 2.89E-09 | 2.59E+02 |
| 1.10E+05 | 1.21E+10 | 24.0919 | 1.99E-09 | 3.14E+02 |
| 1.20E+05 | 1.44E+10 | 35.1938 | 2.44E-09 | 3.74E+02 |
| 1.30E+05 | 1.69E+10 | 30.4317 | 1.80E-09 | 4.39E+02 |
| 1.40E+05 | 1.96E+10 | 30.5835 | 1.56E-09 | 5.09E+02 |
| 1.50E+05 | 2.25E+10 | 24.5055 | 1.09E-09 | 5.84E+02 |
| 1.60E+05 | 2.56E+10 | 34.5979 | 1.35E-09 | 6.64E+02 |
| 1.70E+05 | 2.89E+10 | 32.3879 | 1.12E-09 | 7.50E+02 |
| 1.80E+05 | 3.24E+10 | 42.8913 | 1.32E-09 | 8.41E+02 |
| 1.90E+05 | 3.61E+10 | 41.1736 | 1.14E-09 | 9.37E+02 |
| 2.00E+05 | 4.00E+10 | 44.248 | 1.11E-09 | 1.04E+03 |




**Conclusion:** After conducting the experiment, I concluded that empirical run times are lower than the predicted run times for all three algorithms.

Also, the empirical and theoretical results are consistent.