Name: Darshan Patel

NJIT ID: 31415244

Subject: Data structure and Algorithms (Programming Assignment 2 **Selection Sort**)

OUTPUT & ANALYSIS:

1) **No. of Comparisons:**

	QUICK SORT	QUICK SELECT	LINEAR SEARCH
N=10000	130951	26634	63013
N=100000	1834352	229631	632625
N=1000000	21208122	3563031	6323794

2) **TIME TAKEN BY ALGORITHMS:**

	QUICK SORT	QUICK SELECT	LINEAR SEARCH
N=10000	8.210642 MILLISEC.	3.434425 MILLISEC.	5.46304 MILLISEC.
N=100000	63.819375 MILLISEC.	11.035751 MILLISEC.	28.088498 MILLISEC.
N=1000000	274.293237 MILLISEC.	31.861612 MILLISEC.	83.671842 MILLISEC.

N= 10000

<terminated> SelectionSort [Java Application] C:\Program Files\Java\jre1.8.0_73\bin\javaw.exe (Nov 7, 2017, 2:10:41 PM)

```
  The Size of Input Data, N = 10000

***QUICK SORT***

ALGORITHM (QUICK SORT) : N = 10000     K = 5000     A[5000] = 4882     # Key Comparisons = 130951
TIME TAKEN BY QUICK SORT = 8.210642 Milliseconds

***QUICK SELECT ***

ALGORITHM (QUICK SELECT) : N = 10000     K = 5000     A[5000] = 4882     # Key Comparisons = 26634
TIME TAKEN BY QUICK SELECT = 3.434425 Milliseconds

***LINEAR SELECT***

ALGORITHM (LINEAR SELECT) : N = 10000     K = 5000     A[5000] = 4882     # Key Comparisons = 63013
TIME TAKEN BY LINEAR SELECT = 5.046304 Milliseconds
```

N= 100000

<terminated> SelectionSort [Java Application] C:\Program Files\Java\jre1.8.0_73\bin\javaw.exe (Nov 7, 2017, 2:04:21 PM)

```
  The Size of Input Data, N = 100000

***QUICK SORT***

ALGORITHM (QUICK SORT) : N = 100000     K = 50000     A[50000] = 49878     # Key Comparisons = 1834352
TIME TAKEN BY QUICK SORT = 63.819375 Milliseconds

***QUICK SELECT ***

ALGORITHM (QUICK SELECT) : N = 100000     K = 50000     A[50000] = 49878     # Key Comparisons = 229631
TIME TAKEN BY QUICK SELECT = 11.035751 Milliseconds

***LINEAR SELECT***

ALGORITHM (LINEAR SELECT) : N = 100000     K = 50000     A[50000] = 49878     # Key Comparisons = 632625
TIME TAKEN BY LINEAR SELECT = 28.088498 Milliseconds
```

N=1000000

```
Problems  Documentation  Properties  Console ☒  Snippets
<terminated> SelectionSort [Java Application] C:\Program Files\Java\jre1.8.0_73\bin\javaw.exe (Nov 7, 2017, 2:05:09 PM)
 The Size of Input Data, N = 1000000

***QUICK SORT***

ALGORITHM (QUICK SORT) : N = 1000000    K = 500000    A[500000] = 499915    # Key Comparisons = 21208122
TIME TAKEN BY QUICK SORT = 274.293237 Milliseconds

***QUICK SELECT ***

ALGORITHM (QUICK SELECT) : N = 1000000    K = 500000    A[500000] = 499915    # Key Comparisons = 3563031
TIME TAKEN BY QUICK SELECT = 31.861612 Milliseconds

***LINEAR SELECT***

ALGORITHM (LINEAR SELECT) : N = 1000000    K = 500000    A[500000] = 499915    # Key Comparisons = 6323794
TIME TAKEN BY LINEAR SELECT = 83.671842 Milliseconds
```

COMPARISON AND ANALYSIS

Average Case Analysis:

Case1 N = 10000
1) Quick Sort Algorithm:

The number of key comparisons is **130951**
Average case complexity of Quick sort is given by,
T(n)<=O(n*logn)<=A*(n*logn)
N=10000, therefore n*logn = **132877**.
As the Number of key comparisons are closer to T(n) by some constant factor **'A'**,
O(n*logn).
Hence,
T(n)= 130951<=A*(132877)
Where A = 0.9855

2) Quick Select –

The number of key comparisons =**26634**
And the average case complexity of Quick select is given by,
T(n)<=O(n)<=B*(n)
Therefore n = **10000**.
As the Number of key comparisons are closer to T(n) by some constant factor **'B'**, that is **O(n)**. Hence,
T(n)= 26634 <=B*(10000)
Where B = 2.6634

3) Linear Select:
The number of key comparisons = **63013**.
And the average case complexity of Linear select is given by,
T(n)<=O(n)<=C*(n)
Therefore n = **10000**.
As the Number of key comparisons are closer to T(n) by some constant factor **'C'**, **O(n)**. Hence,
T(n)= 63013<=C*(10000)
Where C = 6.3013

Case-2- N = 100,000
1) Quick Sort:
According to the Tabulated data,
The number of key comparisons = **1834352**.
And the average case complexity of Quick sort is given by,
T(n)<=O(n*logn)<=B*(n*logn)
Here n=100000, therefore n*logn = **1660964**.
As the Number of key comparisons are closer to T(n) by some constant factor **'A'**, **O(n*logn)**. Hence,
T(n)= 1834352<=B*(1660964)
Where A = 1.04389

2) Quick Select –

According to the Tabulated data, the number of key comparisons = **229631**.
And the average case complexity of Quick select is given by,

T(n)<=O(n)<=B*(n)
Therefore n = **100000**.
As the Number of key comparisons are closer to T(n) by some constant factor **'B'**,
i.e. O(n). Hence,
T(n)= 229631<=B*(100000)
Where B = 2.29631

3) Linear Select :

According to the Tabulated data, the number of key comparisons = **632625**.
And the average case complexity of Linear select is given by,
T(n)<=O(n)<=C*(n)
Therefore n = **100000**.
As the Number of key comparisons are closer to T(n) by some constant factor **'C'**,
i.e. O(n). Hence,
T(n)= 632625<=C*(100000)
Where C = 6.32625

Case-3- N = 1,000,000

1) Quick Sort:
According to the Tabulated data,
The number of key comparisons = **21208122**.
And the average case complexity of Quick sort is given by,
T(n)<=O(n*logn)<=A*(n*logn)
Here n=1000000, therefore n*logn = **19931568**.
As the Number of key comparisons are closer to T(n) by some constant factor **'A'**,
O(n*logn). Hence,
T(n)= 21208122<=A*(19931568)
Where A = 1.064

2) Quick Select:
The number of key comparisons = **3563031**.
And the average case complexity of Quick select is given by,
T(n)<=O(n)<=B*(n)
Therefore n = **1000000**.

As the Number of key comparisons are closer to T(n) by some constant factor **'B'**, **i.e. O(n)**. Hence,

T(n)= 3563031<=B*(1000000)

Where B = 3.563031

3) Linear Select :

According to the Tabulated data,

The number of key comparisons = **6323794**.

And the average case complexity of Linear select is given by,

T(n)<=O(n)<=C*(n)

Therefore n = **1000000**.

As the Number of key comparisons are closer to T(n) by some constant factor **'C'**, **i.e. O(n)**. Hence,

T(n)= 6323794<=C*(1000000)

Where C = 6.323794

Concluding Discussion:

Here all complexity is in terms of an average case complexity. Quick sort has a higher complexity as it first sorts the integers array list and then finds the Kth smallest and hence it takes more time than other algorithms. The Quick select algorithm will randomly select a pivot and place that pivot at its correct position and as soon as it finds the required Kth smallest element it returns. The Linear select algorithm finds the Kth smallest element in the average time with complexity of O(n) but constant is higest.so it takes more time than Quick select algorithm.