NAME: OSABHIE EBENEZER FERGUSON
COURSE NUMNER: CSCI 7432
PROGRAMMING ASSIGNMENT NO.: 1
DATE: 30TH JANUARY, 2022

DESCRIPTION: This program sorts an array of random numbers (integers) within a range of 100 and 1000 using the Randomized Quicksort algorithm. A test was carried out for 100, 1000, 10 000 and 50 000 random numbers between 100 and 1000 inclusive.

It compares two different partitioning approaches (Lomuto and Hoare). The program compares performance of the both using two main factors:

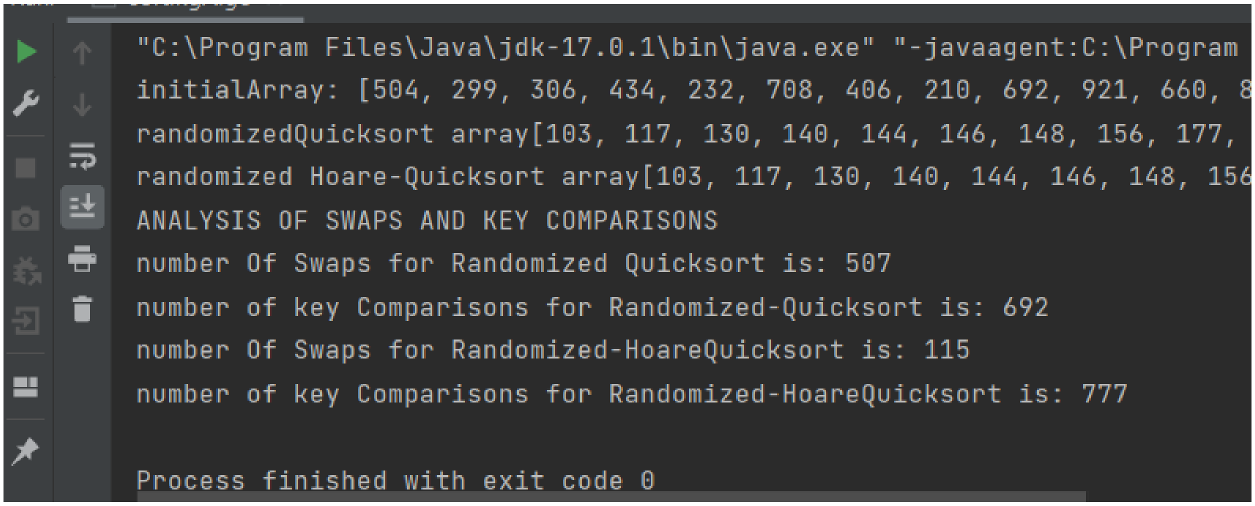
1. Number of Swaps.

2. Number of key comparisons.

HOW TO COMPILE AND RUN: Open the .java file in any suitable Integrated Development Environment (IDE) of your choice. You must have the Java Development Kit (JDK) which contains the Java Virtual Machine (JVM) installed on your machine for the program to run. Click run in the IDE and the code will compile and run.

TESTS

FOR AN ARRAY OF 100 ELEMENTS WITHIN A RANGE OF 100 – 1000



```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-javaagent:C:\Program
initialArray: [504, 299, 306, 434, 232, 708, 406, 210, 692, 921, 660, 8
randomizedQuicksort array[103, 117, 130, 140, 144, 146, 148, 156, 177,
randomized Hoare-Quicksort array[103, 117, 130, 140, 144, 146, 148, 156
ANALYSIS OF SWAPS AND KEY COMPARISONS
number Of Swaps for Randomized Quicksort is: 507
number of key Comparisons for Randomized-Quicksort is: 692
number Of Swaps for Randomized-HoareQuicksort is: 115
number of key Comparisons for Randomized-HoareQuicksort is: 777

Process finished with exit code 0
```

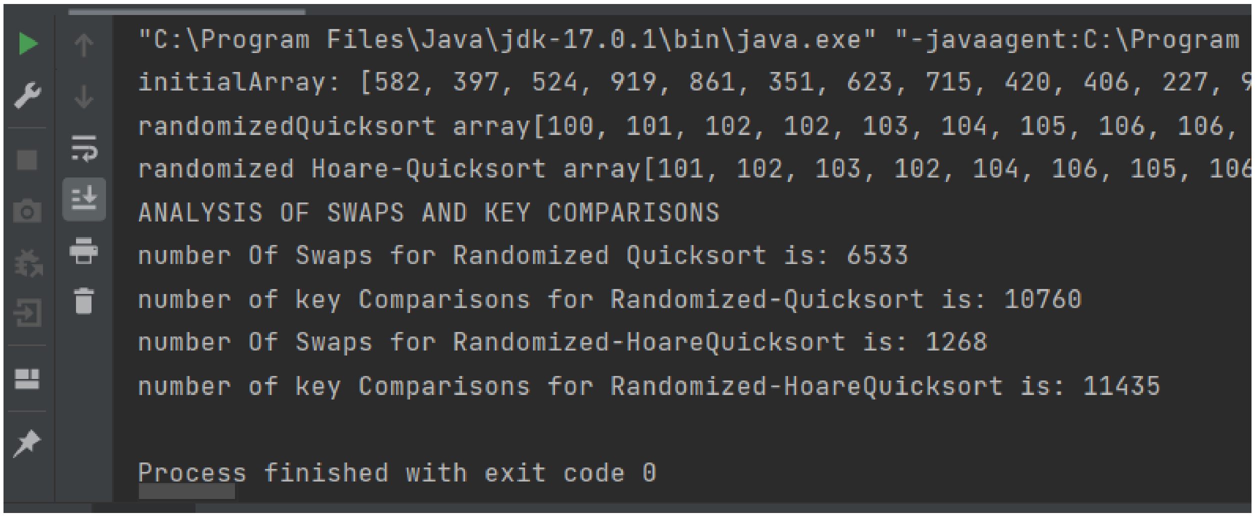
After multiple runs, the image above shows the average results for the test carried out for an array of 100 elements (100 - 1000).

LOMUTO PARTITION (RANDOMIZED QUICKSORT): ~507 Swaps and ~ 692 key comparisons.

RANDOMIZED HOARE-PARTITION: ~115 swaps and ~ 777 key comparisons

AVERAGE RESULTS: HOARE has 4.4 times less swaps than LOMUTO and 1.1 more key comparisons than LOMUTO.

<u>FOR AN ARRAY OF 1,000 ELEMENTS WITHIN A RANGE OF 100 – 1000</u>



```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-javaagent:C:\Program
initialArray: [582, 397, 524, 919, 861, 351, 623, 715, 420, 406, 227, 9
randomizedQuicksort array[100, 101, 102, 102, 103, 104, 105, 106, 106,
randomized Hoare-Quicksort array[101, 102, 103, 102, 104, 106, 105, 106
ANALYSIS OF SWAPS AND KEY COMPARISONS
number Of Swaps for Randomized Quicksort is: 6533
number of key Comparisons for Randomized-Quicksort is: 10760
number Of Swaps for Randomized-HoareQuicksort is: 1268
number of key Comparisons for Randomized-HoareQuicksort is: 11435

Process finished with exit code 0
```

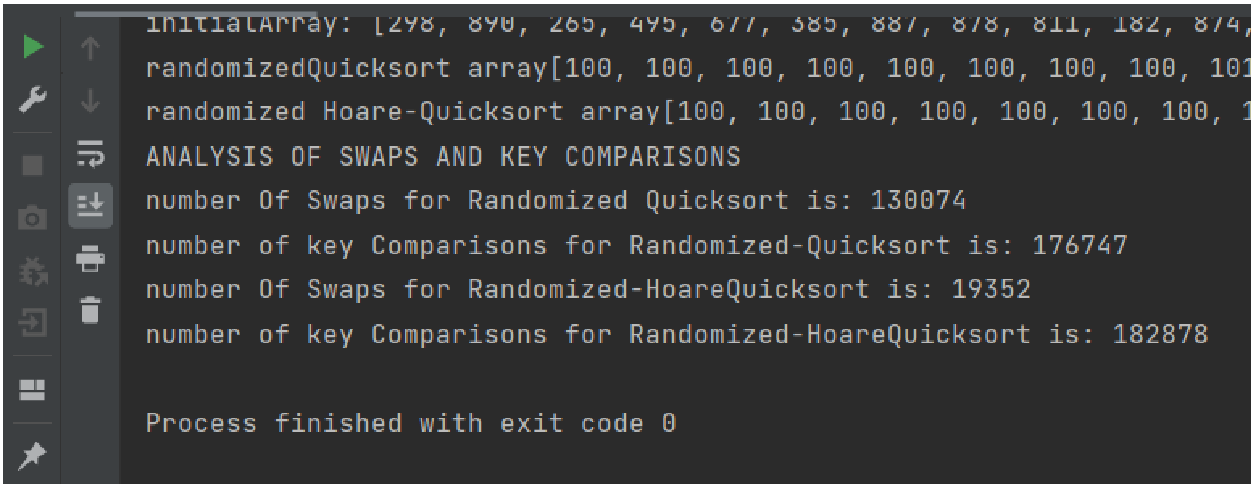
After multiple runs, the image above shows the average results for the test carried out for an array of 1000 elements (100 - 1000).

LOMUTO PARTITION (RANDOMIZED QUICKSORT): ~6533 Swaps and ~ 10760 key comparisons.

RANDOMIZED HOARE-PARTITION: ~1268 swaps and ~ 11,435 key comparisons.

AVERAGE RESULTS: HOARE has 5.2 times less swaps than LOMUTO and 1.1 more key comparisons than LOMUTO.

FOR AN ARRAY OF 10,000 ELEMENTS WITHIN A RANGE OF 100 – 1000



```
initialArray: [298, 890, 265, 495, 677, 385, 887, 878, 811, 182, 874,
randomizedQuicksort array[100, 100, 100, 100, 100, 100, 100, 100, 101
randomized Hoare-Quicksort array[100, 100, 100, 100, 100, 100, 100, 1
ANALYSIS OF SWAPS AND KEY COMPARISONS
number Of Swaps for Randomized Quicksort is: 130074
number of key Comparisons for Randomized-Quicksort is: 176747
number Of Swaps for Randomized-HoareQuicksort is: 19352
number of key Comparisons for Randomized-HoareQuicksort is: 182878

Process finished with exit code 0
```

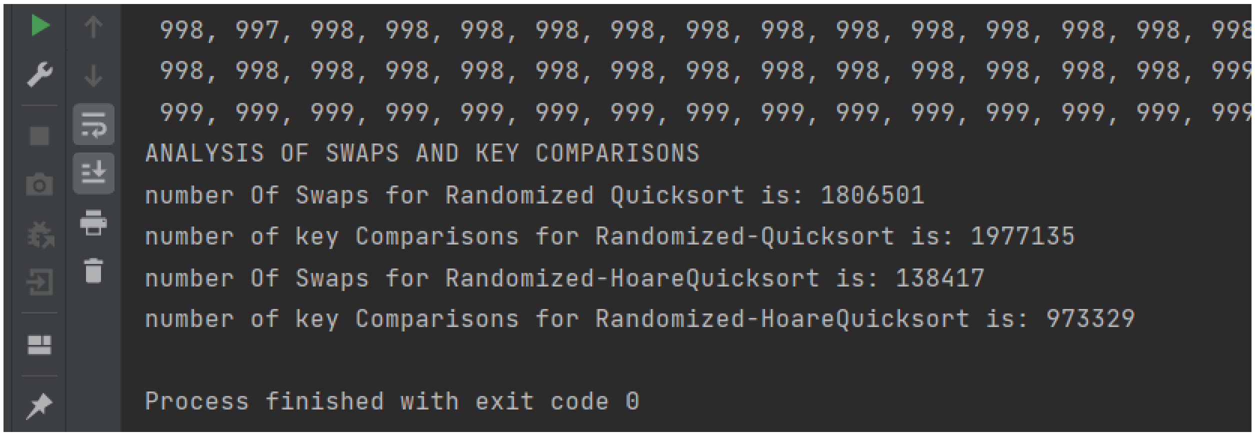
After multiple runs, the image above shows the average results for the test carried out for an array of 10,000 elements (100 - 1000).

LOMUTO PARTITION (RANDOMIZED QUICKSORT): ~130,074 Swaps and ~ 176,747 key comparisons.

RANDOMIZED HOARE-PARTITION: ~19,352 swaps and ~ 182,878 key comparisons.

AVERAGE RESULTS: HOARE has 6.7 times less swaps than LOMUTO and 1.0 more key comparisons than LOMUTO.

FOR AN ARRAY OF 50,000 ELEMENTS WITHIN A RANGE OF 100 – 1000



```
 998, 997, 998, 998, 998, 998, 998, 998, 998, 998, 998, 998, 998, 998, 998
 998, 998, 998, 998, 998, 998, 998, 998, 998, 998, 998, 998, 998, 998, 999
 999, 999, 999, 999, 999, 999, 999, 999, 999, 999, 999, 999, 999, 999, 999
ANALYSIS OF SWAPS AND KEY COMPARISONS
number Of Swaps for Randomized Quicksort is: 1806501
number of key Comparisons for Randomized-Quicksort is: 1977135
number Of Swaps for Randomized-HoareQuicksort is: 138417
number of key Comparisons for Randomized-HoareQuicksort is: 973329

Process finished with exit code 0
```

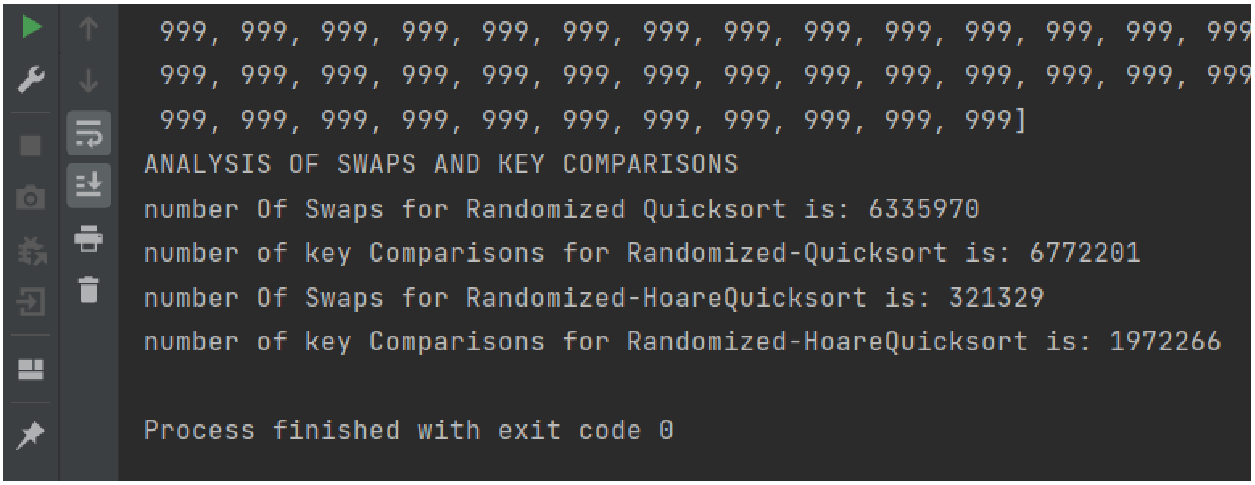
After multiple runs, the image above shows the average results for the test carried out for an array of 50,000 elements (100 - 1000).

LOMUTO PARTITION (RANDOMIZED QUICKSORT): ~1,806,501 Swaps and ~ 1,977,135 key comparisons.

RANDOMIZED HOARE-PARTITION: ~138,417 swaps and ~ 973,329 key comparisons.

AVERAGE RESULTS: HOARE has 13 times less swaps than LOMUTO and 2.0 less key comparisons than LOMUTO.

<u>FOR AN ARRAY OF 100,000 ELEMENTS WITHIN A RANGE OF 100 – 1000</u>



```
 999, 999, 999, 999, 999, 999, 999, 999, 999, 999, 999, 999, 999, 999
 999, 999, 999, 999, 999, 999, 999, 999, 999, 999, 999, 999, 999, 999
 999, 999, 999, 999, 999, 999, 999, 999, 999, 999, 999]
ANALYSIS OF SWAPS AND KEY COMPARISONS
number Of Swaps for Randomized Quicksort is: 6335970
number of key Comparisons for Randomized-Quicksort is: 6772201
number Of Swaps for Randomized-HoareQuicksort is: 321329
number of key Comparisons for Randomized-HoareQuicksort is: 1972266

Process finished with exit code 0
```

After multiple runs, the image above shows the average results for the test carried out for an array of 100,000 elements (100 - 1000).

LOMUTO PARTITION (RANDOMIZED QUICKSORT): ~6,335,970 Swaps and ~ 6,772,201 key comparisons.

RANDOMIZED HOARE-PARTITION: ~321,329 swaps and ~ 1,972,266 key comparisons.

AVERAGE RESULTS: HOARE has 19.7 times less swaps than LOMUTO and 3.4 less key comparisons than LOMUTO.

SUMMARY: Based on the tests above, it is evident that the HOARE partition is a better algorithm than the LOMUTO partition.

	RANDOMIZED-LOMUTO	**RANDOMIZED-HOARE**
Implementation	Way easier to implement	More complex to implement

Swaps	Has way higher number of swaps compared to the HOARE partition that increases even more as the number of elements in the array increases.	Has a lesser number of swaps compared to the LOMUTO partition. Swap difference gets more conspicuous as number of elements in array increases.
Key Comparisons	Number of key comparisons is slightly lower than that of the HOARE partition for a smaller number of elements in array, but becomes more than that of the HOARE partition for a larger array.	Slightly larger of key comparisons than LOMUTO for smaller arrays. As the array increases, key comparisons become way smaller than that of the LOMUTO.

In summary, the **RANDOMIZED-HOARE QUICKSORT** algorithm is way more efficient than the **RANDOMIZED QUICKSORT (LOMUTO PARTITION)** algorithm, as it has less number of swaps and proves to be more efficient as size of the array scales.