Name: Patel Darshan Atulbhai

Njit id: dap83

Program 1 (Output & Analysis)

Teacher: David Nassimi

1)Where N=32

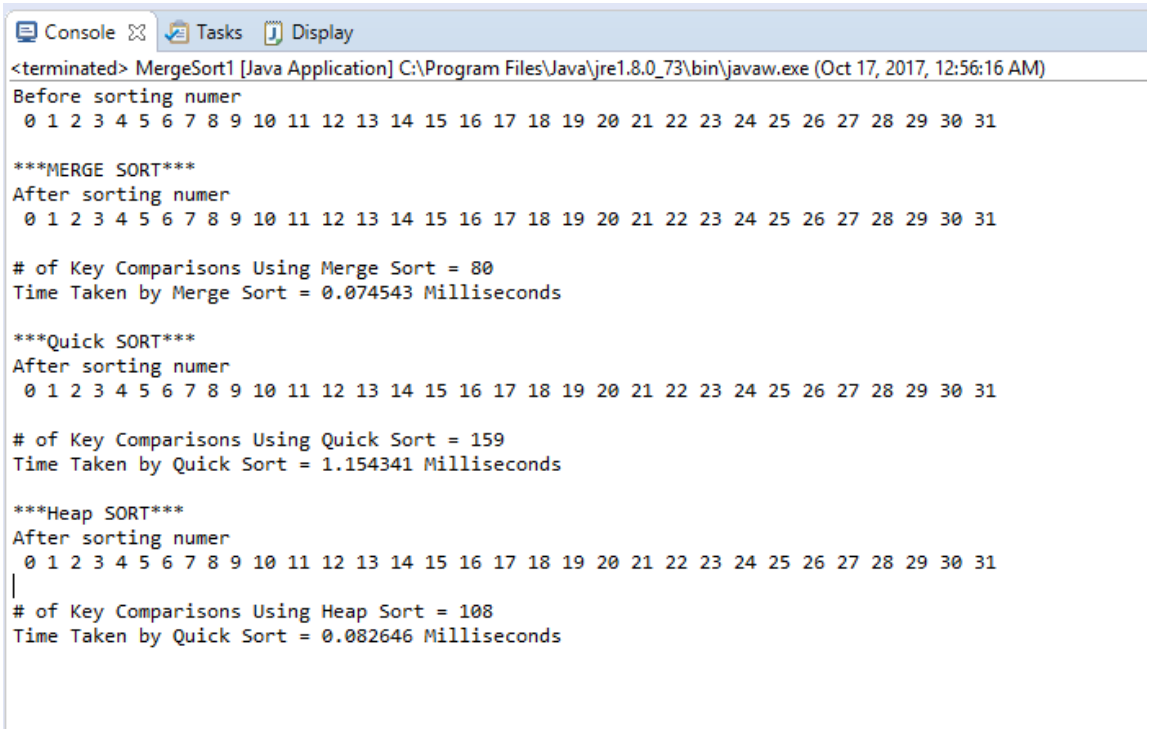
- **No of Key Comparison:**

	Merge Sort	Quick Sort	Heap Sort
Best Case	80	159	108
Avg. Case	124	142	115
Worst Case	80	162	108

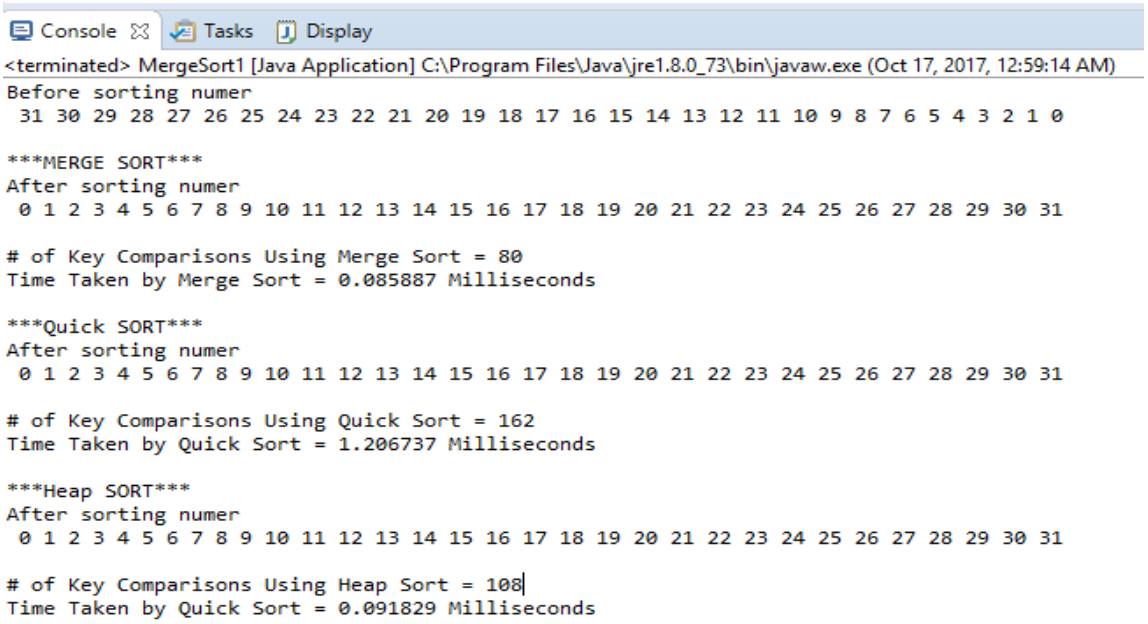
- **Time in Millisecond:**

	Merge Sort	Quick Sort	Heap Sort
Best Case	0.074543	1.154341	0.082646
Avg. Case	0.086967	0.358132	0.111815
Worst Case	0.085887	1.206737	0.091829

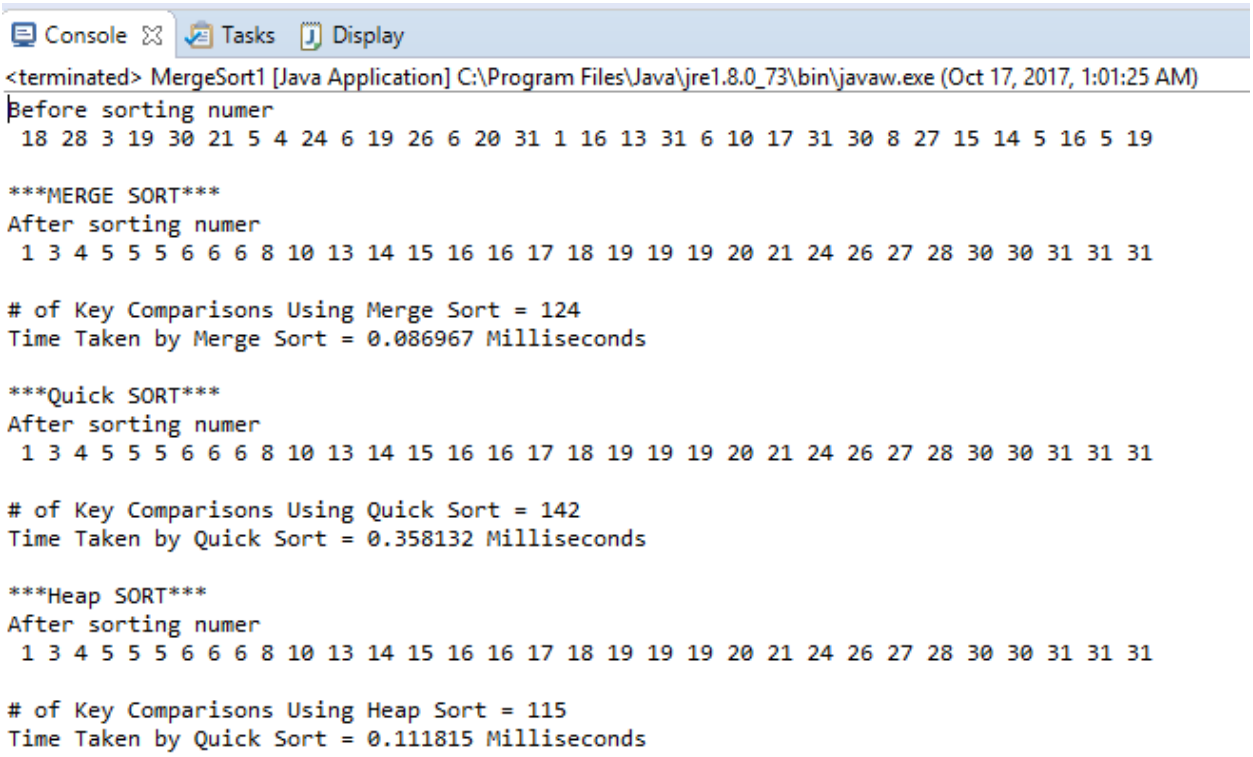
Best Case:



Worst Case:



AverageCase:



2)Large Size Array:

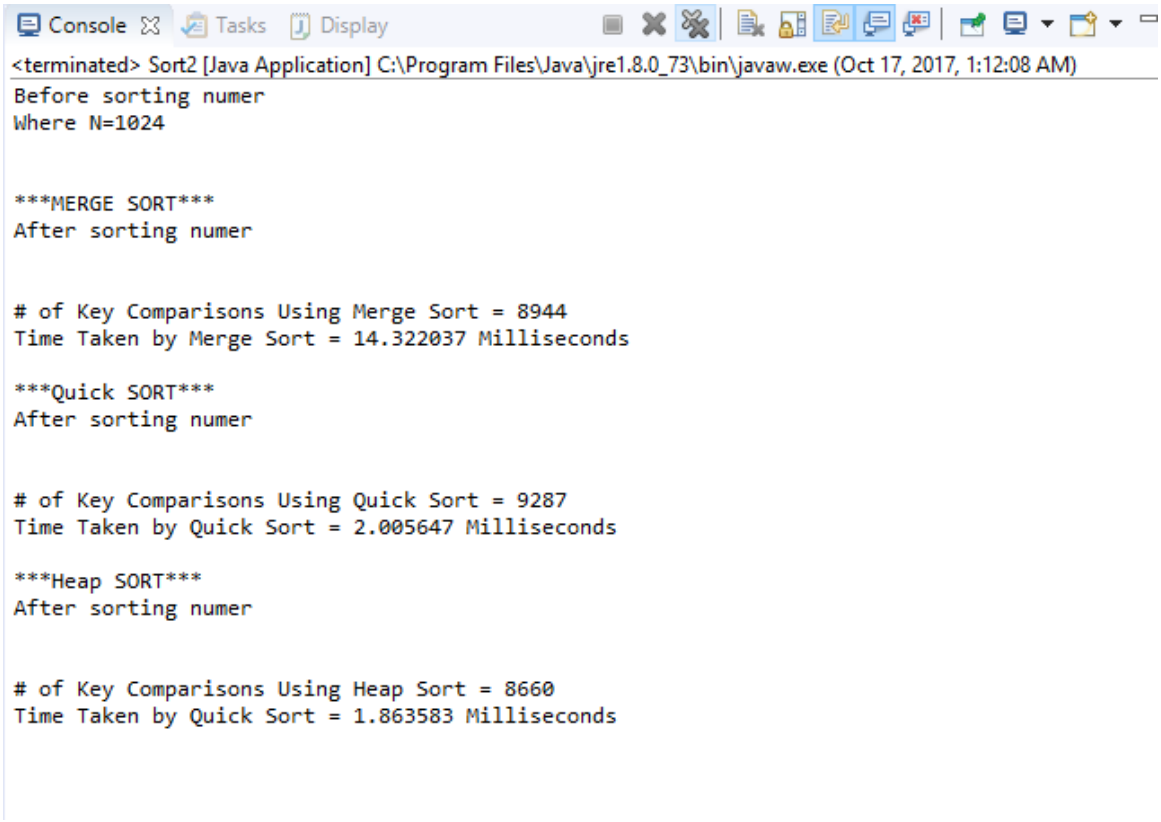
- **No of key Comparison:**

N	Merge Sort	Quick Sort	Heap Sort
1024	8944	9287	8660
32768	450017	516480	440822
1048576	19644941	22047670	19352696

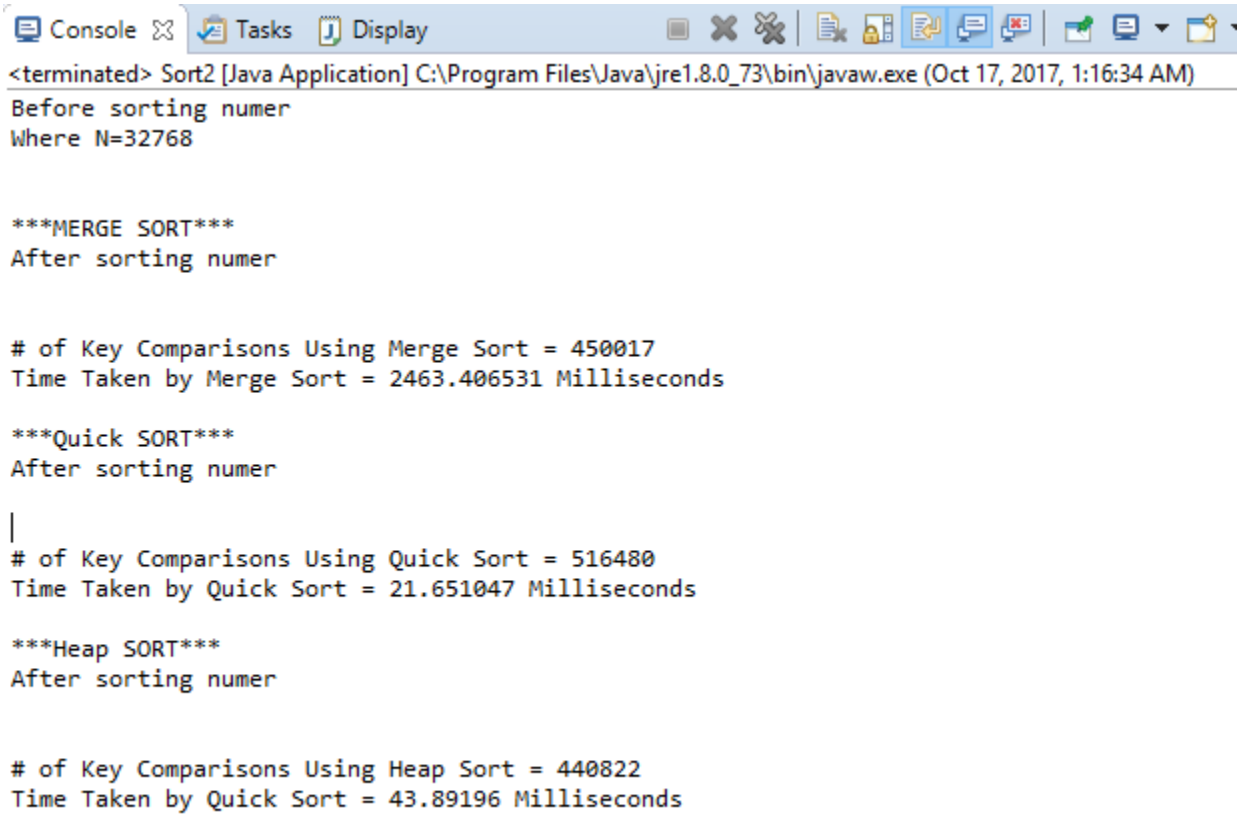
- **Time in Millisecond:**

N	Merge Sort	Quick Sort	Heap Sort
1024	14.322037	2.005647	1.863583
32768	2463.406531	21.651047	43.89196
1048576	2025794.820645	557.452629	519.865519

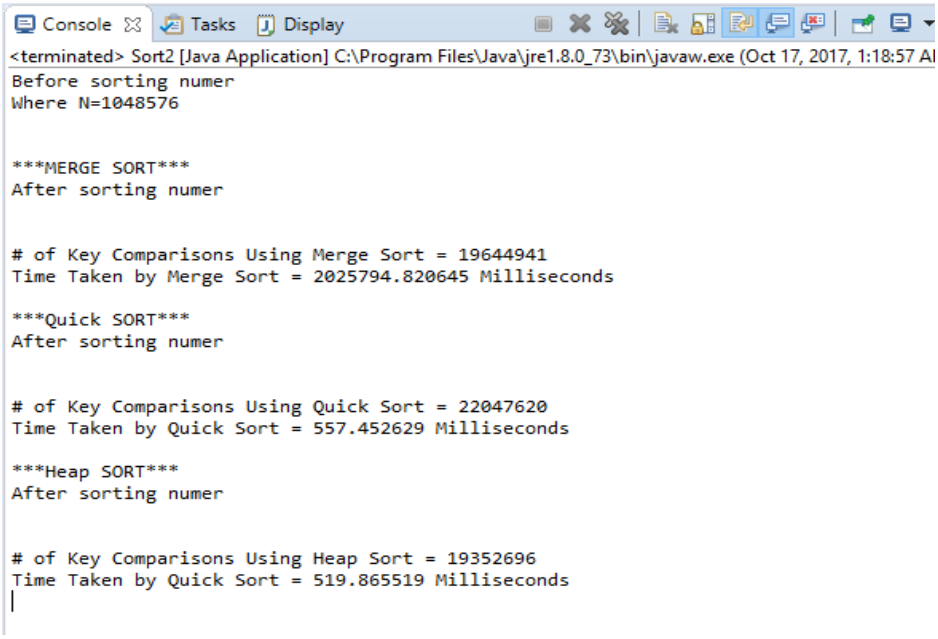
N=1024



N=32768


Console
Tasks
Display
<terminated> Sort2 [Java Application] C:\Program Files\Java\jre1.8.0_73\bin\javaw.exe (Oct 17, 2017, 1:16:34 AM)
Before sorting numer
Where N=32768
MERGE SORT
After sorting numer
# of Key Comparisons Using Merge Sort = 450017
Time Taken by Merge Sort = 2463.406531 Milliseconds
Quick SORT
After sorting numer
# of Key Comparisons Using Quick Sort = 516480
Time Taken by Quick Sort = 21.651047 Milliseconds
Heap SORT
After sorting numer
# of Key Comparisons Using Heap Sort = 440822
Time Taken by Quick Sort = 43.89196 Milliseconds

N=1048576


Console
Tasks
Display
<terminated> Sort2 [Java Application] C:\Program Files\Java\jre1.8.0_73\bin\javaw.exe (Oct 17, 2017, 1:18:57 AM
Before sorting numer
Where N=1048576
MERGE SORT
After sorting numer
# of Key Comparisons Using Merge Sort = 19644941
Time Taken by Merge Sort = 2025794.820645 Milliseconds
Quick SORT
After sorting numer
# of Key Comparisons Using Quick Sort = 22047620
Time Taken by Quick Sort = 557.452629 Milliseconds
Heap SORT
After sorting numer
# of Key Comparisons Using Heap Sort = 19352696
Time Taken by Quick Sort = 519.865519 Milliseconds

Analysis: Here we take integer as random number so it is nothing but average case analysis.

Case-1: N = 1024

1) Merge Sort :

According to the Tabulated data, the number of key comparisons = 8944.

And the average case complexity of Merge sort is given by,

T(n)<=O(n*logn)<=A*(n*logn)

Here n=1024, therefore n*logn = 10240.

As the Number of key comparisons are closer to T(n) by some constant factor 'A',

So the tabulated data corresponds to the average case analysis of Merge sort i.e. O(n*logn). Hence,

T(n)= 8944<=A*(10240)

Where A = 0.873

2) Quick Sort:

According to the Tabulated data, the number of key comparisons = 9287.

And the average case complexity of Quick sort is given by,

T(n)<=O(n*logn)<=B*(n*logn)

Here n=1024, therefore n*logn = 10240.

As the Number of key comparisons are closer to T(n) by some constant factor 'B',

So the tabulated data corresponds to the average case analysis of Quick sort i.e. O(n*logn). Hence,

T(n)= 9287<=B*(10240)

Where B = 0.907

3) Heap Sort:

According to the Tabulated data, the number of key comparisons = 8660.

And the average case complexity of Heap sort is given by,

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

Here n=1024, therefore n*logn = 10240.

As the Number of key comparisons are closer to T(n) by some constant factor ‘C’,

So the tabulated data corresponds to the average case analysis of Heap sort i.e.

O(n*logn). Hence,

T(n)= 8660<=C*(10240)

Where C = 0.845

Case-2: N = 32768

1) Merge Sort:

According to the Tabulated data, the number of key comparisons = 450017.

And the average case complexity of Merge sort is given by,

T(n)<=O(n*logn)<=A*(n*logn)

Here n=32768, therefore n*logn = 491520.

As the Number of key comparisons are closer to T(n) by some constant factor ‘A’,

So the tabulated data corresponds to the average case analysis of Merge sort i.e.

O(n*logn). Hence,

T(n)= 450017<=A*(491520)

Where A = 0.915

2) Quick Sort:

According to the Tabulated data, the number of key comparisons = 516480.

And the average case complexity of Quick sort is given by,

T(n)<=O(n*logn)<=B*(n*logn)

Here n=32768, therefore n*logn = 491520.

As the Number of key comparisons are closer to T(n) by some constant factor ‘B’,

So the tabulated data corresponds to the average case analysis of Quick sort i.e.

O(n*logn). Hence,

T(n)=516480<=B*(491520)

Where B = 1.050

3) Heap Sort:

According to the Tabulated data, the number of key comparisons = 440822.

And the average case complexity of Heap sort is given by,

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

Here n=32768, therefore n*logn = 491520.

As the Number of key comparisons are closer to T(n) by some constant factor ‘C’,

So the tabulated data corresponds to the average case analysis of Heap sort i.e.

O(n*logn). Hence,

T(n)= 440822<=C*(491520)

Where C = 0.896

Case-3: N = 1048576

1) Merge Sort:

According to the Tabulated data,

The number of key comparisons = 19644941.

And the average case complexity of Merge sort is given by,

T(n)<=O(n*logn)<=A*(n*logn)

Here n=1048576, therefore n*logn = 20971520.

As the Number of key comparisons are closer to T(n) by some constant factor ‘A’, So the tabulated data corresponds to the average case analysis of Merge sort i.e. O(n*logn). Hence,

T(n)= 19644941<=A*(20971520)

Where A = 0.937

2) Quick Sort:

According to the Tabulated data,

The number of key comparisons = 22047670.

And the average case complexity of Quick sort is given by,

T(n)<=O(n*logn)<=B*(n*logn)

Here n=1048576, therefore n*logn = 20971520.

As the Number of key comparisons are closer to T(n) by some constant factor ‘B’, So the tabulated data corresponds to the average case analysis of Quick sort i.e. O(n*logn). Hence,

T(n)= 22047670<=B*(20971520)

Where B = 1.0513

3) Heap Sort:

According to the Tabulated data,

The number of key comparisons =19352696.

And the average case complexity of Heap sort is given by,

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

Here n=1048576, therefore n*logn = 20971520.

As the Number of key comparisons are closer to T(n) by some constant factor ‘C’, So the tabulated data corresponds to the average case analysis of Heap sort i.e. O(n*logn). Hence,

T(n)= 19352696<=C*(20971520)

Where C = 0.923