Name	Kanchan Waikar
UTD ID	2021255617
Project	CS 6312.002 : Short Project 0

Description:

In merge Sort algorithm, The merge function algorithm was slightly modified to use in place sorting using a single temporary array. This improved the space complexity of Merge sort. For Implementing Heapsort, Priority Queue implementation was used. While testing it was found that whenever we use method to sort, it does not do in-place sorting and thus needs to explicitly return data. When we do so, the performance goes down. Hence both statistics were provided for the same.

Based on following statistics we can see that average time taken by Merge sort is much better than Heapsort although both are nlogn complexity algorithms.

Statistics:
Data Size: **1000001**

Type of Input	**Merge Sort Statistics**	**Sorting using Priority Queue without printing top 10 entries**	**Sorting using Priority Queue and priting top 10 entries**
Integer	Average Time taken by 3 Trials = 380 msec Average Memory taken by 3 Trials = 70 MB	Average Time taken by 5 Trials = 632 msec Average Memory taken by 5 Trials = 32 MB	Average Time taken by 3 Trials = 728 msec Average Memory taken by 3 Trials = 59 MB

Data Size: **2000001**

Type of Input	**Merge Sort Statistics**	**HeapSort using Priority Queue without printing top 10 entries**	**HeapSort that prints top 10 entries**
Integer	Average Time taken by 3 Trials = 720 msec Average Memory taken by 3	Average Time taken by 3 Trials = 1441 msec Average Memory taken by 3	Average Time taken by 3 Trials = 1681 msec Average Memory taken by 3

	Trials = 70 MB	Trials = 48 MB	Trials = 101 MB
Float	Average Time taken by 3 Trials = 668 msec Average Memory taken by 3 Trials = 70 MB	Average Time taken by 3 Trials = 1463 msec Average Memory taken by 3 Trials = 48 MB	Average Time taken by 5 Trials = 1725 msec Average Memory taken by 5 Trials = 102 MB
Double	Average Time taken by 3 Trials = 763 msec Average Memory taken by 3 Trials = 116 MB	Average Time taken by 5 Trials = 1510 msec Average Memory taken by 5 Trials = 65 MB	Average Time taken by 5 Trials = 1798 msec Average Memory taken by 5 Trials = 117 MB

How to Execute The program

1. Explode the zip file in a folder
2. Go inside above folder
3. execute command “Javac *.java” - This compiles all three programs
4. Execute command “java SortDriver” This would execute the Program and would ask you to provide following inputs
    a. Number of trials you want to execute
    b. Number of random integers you want to sort
    c. Type of algorithm you want to choose for sorting
5. Once done, the program executes it displays consolidated results as shown below. It allows you to choose any of the two algorithms.

Execution log:

```
P:\workspace\Project_0 Sorting Algorithm comparison\src>java SortDriver
Please provide number of trials you would like to execute:3
Please provide number of random integers you would like to load in the input data:2000001
Please select Sorting technique
1: Merge Sort
2: Sorting using Priority Queue
Enter your choice:1

List/Array before sorting  : 269042 392768 679754 908169 734381 291845 653415 23034 275589 94136
Time: 805 msec.
Memory: 83 MB / 243 MB.
----------------------------------------
```

List/Array after sorting : 0 3 3 3 3 4 4 5 5 5
After Sorting

List/Array before sorting : 594536 636699 620509 356866 693085 56604 988026 332080 9277 853203
Time: 698 msec.
Memory: 68 MB / 292 MB.

List/Array after sorting : 0 0 2 5 5 5 7 7 8 9
After Sorting

List/Array before sorting : 957706 611104 65880 93004 569270 946577 446931 42420 394676 972460
Time: 659 msec.
Memory: 61 MB / 294 MB.

List/Array after sorting : 0 1 1 1 3 3 5 6 7 7
After Sorting
==========
Average Time taken by 3 Trials = 720 msec
Average Memory taken by 3 Trials = 70 MB