```
Time required to sort random arr using MergeSort(Recursive) with 10^4 elements: 0.004247 sec
Time required to sort sorted arr using MergeSort(Recursive) with 10^4 elements: 0.002282 sec
Time required to sort swapped arr using MergeSort(Recursive) with 10^4 elements: 0.00236 sec

Time required to sort random arr using MergeSort(Recursive) with 10^5 elements: 0.039577 sec
Time required to sort sorted arr using MergeSort(Recursive) with 10^5 elements: 0.025257 sec
Time required to sort swapped arr using MergeSort(Recursive) with 10^5 elements: 0.025637 sec

Time required to sort random arr using MergeSort(Recursive) with 10^6 elements: 0.463521 sec
Time required to sort sorted arr using MergeSort(Recursive) with 10^6 elements: 0.285351 sec
Time required to sort swapped arr using MergeSort(Recursive) with 10^6 elements: 0.289845 sec

Time required to sort random arr using MergeSort(Recursive) with 10^7 elements: 5.64638 sec
Time required to sort sorted arr using MergeSort(Recursive) with 10^7 elements: 3.13842 sec
Time required to sort swapped arr using MergeSort(Recursive) with 10^7 elements: 3.5863 sec

Time required to sort random arr using MergeSort(Iterative) with 10^4 elements: 0.003333 sec
Time required to sort sorted arr using MergeSort(Iterative) with 10^4 elements: 0.002283 sec
Time required to sort swapped arr using MergeSort(Iterative) with 10^4 elements: 0.002299 sec

Time required to sort random arr using MergeSort(Iterative) with 10^5 elements: 0.038818 sec
Time required to sort sorted arr using MergeSort(Iterative) with 10^5 elements: 0.025379 sec
Time required to sort swapped arr using MergeSort(Iterative) with 10^5 elements: 0.025779 sec

Time required to sort random arr using MergeSort(Iterative) with 10^6 elements: 0.44663 sec
Time required to sort sorted arr using MergeSort(Iterative) with 10^6 elements: 0.281697 sec
Time required to sort swapped arr using MergeSort(Iterative) with 10^6 elements: 0.28653 sec

Time required to sort random arr using MergeSort(Iterative) with 10^7 elements: 5.06879 sec
Time required to sort sorted arr using MergeSort(Iterative) with 10^7 elements: 3.2115 sec
Time required to sort swapped arr using MergeSort(Iterative) with 10^7 elements: 3.27162 sec

The amount of space is used during running the program = Maximum resident set size (kbytes) [ ===>
57th line ]

Large size input data is handled by 'Array' Data Structure of int type where data is read from input
file and output is written back to output file

Memory release statement is present from 71th line onwards

MergeSort takes O(nlogn) time to perfrom sorting operation in all cases as working of Merge sort is
independent of distribution of data

Disadvantage of Merge sort implemented either recursively or non-recursively is that it requires an
extra space of O(n)

==29638== Memcheck, a memory error detector
==29638== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==29638== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==29638== Command: /usr/bin/time -v ./a.out
==29638==
        Command being timed: "./a.out"
        User time (seconds): 28.88
        System time (seconds): 0.44
        Percent of CPU this job got: 99%
        Elapsed (wall clock) time (h:mm:ss or m:ss): 0:29.33
        Average shared text size (kbytes): 0
        Average unshared data size (kbytes): 0
        Average stack size (kbytes): 0
        Average total size (kbytes): 0
        Maximum resident set size (kbytes): 129308
        Average resident set size (kbytes): 0
        Major (requiring I/O) page faults: 0
        Minor (reclaiming a frame) page faults: 102067
        Voluntary context switches: 3
        Involuntary context switches: 164
        Swaps: 0
        File system inputs: 0
        File system outputs: 342528
        Socket messages sent: 0
        Socket messages received: 0
        Signals delivered: 0
        Page size (bytes): 4096
        Exit status: 0
==29638==
==29638== HEAP SUMMARY:
==29638==     in use at exit: 715 bytes in 1 blocks
==29638==   total heap usage: 1 allocs, 0 frees, 715 bytes allocated
==29638==
==29638== 715 bytes in 1 blocks are still reachable in loss record 1 of 1
```

```
==29638==    at 0x483B7F3: malloc (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-
linux.so)
==29638==    by 0x10951B: ??? (in /usr/bin/time)
==29638==    by 0x48830B2: (below main) (libc-start.c:308)
==29638==
==29638== LEAK SUMMARY:
==29638==    definitely lost: 0 bytes in 0 blocks
==29638==    indirectly lost: 0 bytes in 0 blocks
==29638==      possibly lost: 0 bytes in 0 blocks
==29638==    still reachable: 715 bytes in 1 blocks
==29638==         suppressed: 0 bytes in 0 blocks
==29638==
==29638== For lists of detected and suppressed errors, rerun with: -s
==29638== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```