Lab Assignment   01

Write the program to implement a **Merge Sort, Quicksort, Bubble Sort, Selection and Insertion sort** algorithms and comparatively analyze its time complexity.

Design and implement the above approaches. Your program must take a file containing list of integers as input. The size of elements in the list will be the numbers stored in the file. Your program invokes the above approaches for each algorithm to sort the elements in the list using the same set of input parameters. Determine the time taken by each approach of the methods for the same size of inputs data. Store the time taken along with the value of input size used into a file for plotting the graph. Ensure that the size of the list of integer elements must be sufficiently large enough for getting any significant values for comparison.

Using the output data file generated, plot a graph for comparative efficiency analysis of the approach of above sorting methods.

Create execution profile of C program, analyze profiling information and calculate amount of time spent in each routine of your program i.e. compile your code for doing the performance evaluation by using **gprof tool**.

Note: Write your program using modules and multi-file programming approach i.e. your program file divided into multiple files and programs into modules.

How to use gprof tool to produce an execution profile of C

 i)   gcc –pg sampleProgram.c –o myProg   //compile and generate myProg executable object

ii)   ./myProg   //Execute the program

iii)   *gmon.out*   // Check this new file generated automatically in working directory

iv) gprof   myProg gmon.out > output.txt   //profiling information present in output.txt

To measure Execution Time:

```
#include <sys/time.h>
int main(int argc,char **argv)
{ ......
```
// Declaration of timeval variables which will capture the system time
```
long time_start, time_end;
struct timeval tv;
```
// Write all the initialization code here
// Time measurement begins here
```
gettimeofday(&tv,NULL);  /*gets the current system time into variable tv */
time_start = (tv.tv_sec *1e+6) + tv.tv_usec;
```
// calculating start time in seconds
// Write the code to be analyzed here
```
gettimeofday (&tv, NULL);
time_end = (tv.tv_sec *1e+6) + tv.tv_usec;
```
// calculating end time in seconds   //convert time in readable and printable precision
// (time_end-time_start) gives the total execution time
```
return (0);
}
```