CS559: Computer Systems Lab -1

Assignment - 1

Date - 27.07.2018

In this assignment you need to develop a framework for comparing some of the sorting algorithms. This framework needs to be developed using some scripting languages- like sed/awk, Perl, Python, etc. Comparison of various sorting algorithms will be done based on (a) the number of comparisons that it makes, (b) execution time.

One will use the following command in a linux shell to analyze the performance of various sorting techniques

```
$>./testsort  -i <min_size> -x <max_size> -s <step_size> -a <avg> -t [bubble merge heap quick3M quickM] -o <out_filename> -g <plot_script_file_name>
```

- `-i <min_size>` : To specify minimum input size. Default value is 50000.
- `-x <max_size>` : To specify maximum input size. Default value is 100000.
- `-s <step_size>` : To specify the steps in which the number of inputs to sorting algorithms will vary. Default value is 10000.
- `-a <avg>` : A sorting technique will be run `avg` number of times on the same input set and finally average will be taken as an estimate for the performance measure. Default value of `avg` is 1.
- `-o <out_filename>` : Performance measure will be stored in this file. Default filename is `output.txt`.
- `-g <plot_script_file_name>` : You need to generate gnuplot script to plot the output that you have obtained in `output.txt`. Default value is `plot.gnu`.
- `-t [ ]` : User can choose any number of methods out of the following six methods. Default values are - `bubble and merge`.
  - `bubble` - Bubble sort
  - `merge` - Merge sort
  - `heap` - Heap sort
  - `quickM` - Quick sort where the median is chosen as pivot element. You need to find median in `O(n)` time.
  - `quick3M` - Quick sort where the pivot element is selected by finding the median of 3 randomly picked numbers.

Please note the following:

- You need to generate the inputs randomly and the same input needs to be used for all the sorting techniques specified by `-t` option.
- You have to plot both run time as well as number of comparisons made.
- Sorting algorithms are to be implemented using C or C++ or JAVA.
- Please provide makefile to build executable(s) for your C/Java code.
- Timeline - 1st week: Need to implement sorting algorithms, 2nd week - Scripting framework and final submission.