

# TestSort vs 1.0

Saswat Raj

February 4, 2013

# 1 User Manual

## 1.1 Overview

TestSort is a command line utility that compares the sorting time taken by the various sort procedures (currently heapsort,mergesort,bubblesort and quicksort) according to the user input of range and other details and plots them onto a graph using the gnuplot utility for linux.

## 1.2 Dependencies

- Linux
- gcc compiler
- bash shell
- gnuplot utility

## 1.3 System Specifications

The user should add the path of the testsort.sh file to the path variable to access the command from anywhere in the command line. It is necessary that the file has the permissions to read and write data to the system.The *src* folder should be the child of the directory containing the testsort file.The testsort.sh should be executable and so should the files be in the src directory.If the files are not executable use the *chmod* command to make them executable.

## 1.4 User Command

To use the testsort function use the command line in linux environment.If the directory of testsort is already in your path then you can directly provide your specifications by using the command :

```
./testsort -i init -r res -f final -a avg -o outputfile  
          -s [heap,merge,quick,bubble]
```

where:

- init :initial value to start testing from
- res :increment per step of the calculation
- final :the final value of the calculation
- avg :the numeber for tavg calculation
- -s ...:the sorts to be tested

**Note :***The -s option should be mentioned in at the last of the input specifications*

**The Calculation Of Data:** The program iterates over from *init* to *final* incremented by *res* and the total time is calculated for *avg* number of such calculations for given number of fixed data. The data is then sent to *gnuplot* for the corresponding output graph. The output file is stored in *src* folder in the project directory.

## 2 Developer Manual:

### 2.1 Structure of Files and their Uses:

- testsort
- src
  - mergesort.c
  - heapsort.c
  - bubblesort.c
  - quicksort.c
  - merge.sh
  - bubble.sh
  - quick.sh
  - heap.sh
  - plot.sh
  - generate.c

***mergesort.c*:** Implements the mergesort algorithm. This generates merge executable during runtime that takes an argument the number of elements in array to sort.

***heapsort.c*:** Implements the heapsort algorithm. This generates heap executable during runtime that takes an argument the number of elements in array to sort.

***bubblesort.c*:** Implements the bubblesort algorithm. This generates bubble executable during runtime that takes an argument the number of elements in array to sort.

***quicksort.c*:** Implements the quicksort algorithm. This generates quick executable during runtime that takes an argument the number of elements in array to sort.

***merge.sh*:** Returns the average time taken by the merge procedure over the generated number of elements. It takes as arguments the average number of times and the number of values in the array.

***heap.sh*:** Returns the average time taken by the heap procedure over the generated number of elements. It takes as arguments the average number

of times and the number of values in the array.

***bubble.sh***:Returns the average time taken by the bubble procedure over the generated number of elements. It takes as arguments the average number of times and the number of values in the array.

***quick.sh***:Returns the average time taken by the quick procedure over the generated number of elements. It takes as arguments the average number of times and the number of values in the array.

***plot.sh***:Uses the gnuplot to plot the data from output file to gnuplot and displays it on the screen.

***generate.c***:Generates the executable gen which takes as arguments the required random numbers to generate and places them in an output file *temp-file.txt*.

## 2.2 Files Generated during runtime :

- src:merge
- src:heap
- src:quick
- src:bubble
- src:gen (generates random numbers)
- src:*output file*

## 2.3 Testsort:

Testsort uses getopt to parse arguments and the -s argument is processed last by shifting the argument index to the very end. It replaces the default values by the values mentioned in the initial arguments after checking for the validity of the arguments. It then generates the C executables if they are not present and runs according to the instructions specified and generates the output file and the gnuplot

### 3 Sample Working:

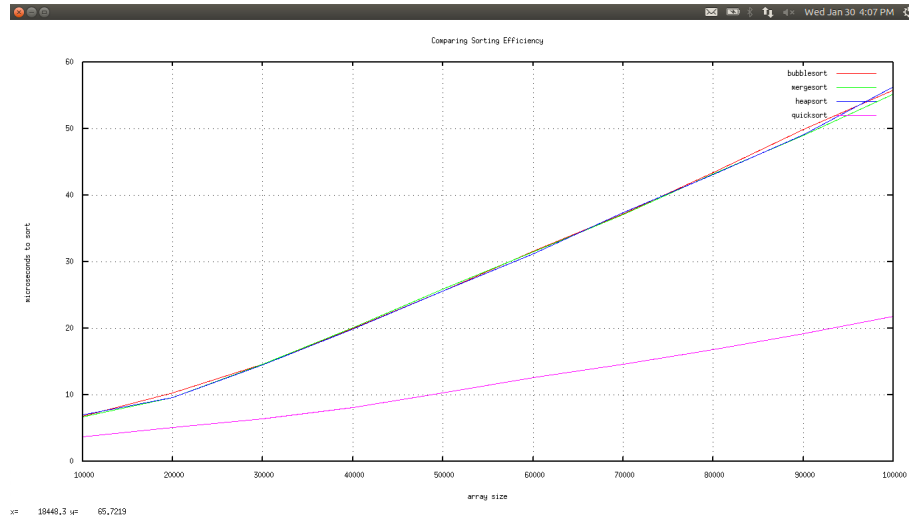


Figure 1: The Graph for larger values when plotted in seconds

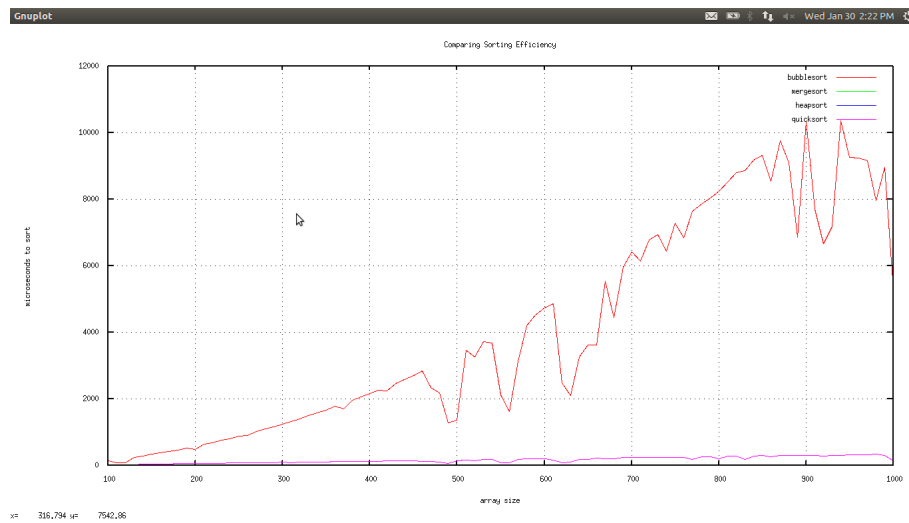


Figure 2: The Graph for smaller values when plotted in microseconds

```
saswat@drakia: ~/Desktop/syslab/assign04
saswat@drakia:~/Desktop/syslab/assign04$ ./testsort --help
usage: testsort -i init [50000] -r res [10000] -f fin [100000] -a avg [10] -o
      filename [out.txt] -s heap merge quick bubble [bubble quick]
      -s argument should be tested last
saswat@drakia:~/Desktop/syslab/assign04$ ./testsort -i 10 -r 10 -f 100 -a 20 -s quick

YOUR SPECIFICATIONS ::
INITIAL VALUE :: 10
INCREMENT VALUE :: 10
FINAL VALUE :: 100
AVERAGE TIMES :: 20
OUTPUT FILE NAME :: out.txt
HEAPSORT TO BE DONE :: false
MERGESORT TO BE DONE :: false
QUICKSORT TO BE DONE :: true
BUBBLESORT TO BE DONE :: false

Please wait while we process the inputs ...
saswat@drakia:~/Desktop/syslab/assign04$
```

Figure 3: Using command in command line