**SNU CSE:** Data Structure | Due Date : 11 pm, Nov 30 2018

**Assignment 4**

Lecturer: Prof. Satti

# Sorting

In this assignment, you should implement a variety of sorting algorithms and compare their performances. Here is the list you should implement:

1. Selection sort
2. Quicksort
3. heapsort, without using `heap initialization' (i.e., by inserting the numbers repeatedly into an initially empty heap)
4. heapsort using heap initialization
5. using a binary search tree (insert the numbers into an initially empty binary search tree, and report the numbers in in-order)
6. using a splay tree (as in 4)
7. merge sort

Write a program that sorts integers by each algorithm and prints the running time (take the average running time after repeating the sorting procedure enough number of times, to improve the accuracy of each. The input data must be read from a text file containing integers separated by white spaces. Write a procedure that verifies that the output produced by a sorting algorithm is indeed sorted.

## Sample Execution

```
[input.txt]

10 18 11 12 13 15 16 19 14 17 ... 10123

$ java Sort input.txt

x1 ms selection sort

x2 ms quicksort

x3 ms heapsort (without heap initialization)

x4 ms heapsort

x5 ms BST sort
```

`x6 ms splay tree sort`

`x7 ms merge sort`

`the sorted file is “output.txt”.`

// (where x1, x2, x3, x4, x5,x6, and x7 are the times, in milliseconds, taken by the corresponding sorting algorithms for the given input).

You should experiment by changing the number of integers and analyze the result. To make these numbers non-zero for a small number of integers, repeat the sorting procedure 100 times on the given input. Submit your report, justifying the above run-times, as a pdf file, along with your program and `output.txt`.

[Create the input file, `input.txt`, by generating a sequence of random numbers of the required length -- say, three different files containing 100, 1000 and 10000 integers.]

# Assessment

Your code will be judged on its correctness. Please make sure that your code runs on Linux terminal. TA will not accept any complain if your code doesn’t run on Linux terminal correctly, even if it runs on other IDEs or working environment correctly.

# Submission

Compress all your source and both input and output files (your code along with its executable file and your result description(pdf)) into a single file as your-full-name_SID.zip.